AR

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 802 492 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
22.10.1997 Bulletin 1997/43

(51) Int Cl.[6]: G06F 17/30

(21) Application number: 97302600.8

(22) Date of filing: 16.04.1997

(84) Designated Contracting States:
DE FR GB

(30) Priority: 17.04.1996 JP 95691/96

(71) Applicant: INTERNATIONAL BUSINESS MACHINES CORPORATION
Armonk, NY 10504 (US)

(72) Inventor: Kubota, Rie
Yamato-shi, Kanagawa-ken (JP)

(74) Representative: Davies, Simon Robert
I B M
UK Intellectual Property Department
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) Document search system

(57) A unique character string is extracted from an input document 907, and a similarity search is performed by using the unique character string. The extraction of the unique character string is performed by calculating and evaluating the amount of feature of a character string through comparison between appearance frequency appearing in the input document 907 and appearance frequency in a set of documents 909 to be searched. Then, the extracted unique character string is used for the search. Documents found by the search are evaluated and arranged in the order of evaluation. The similarity factor of document is evaluated by using the appearance frequency of each unique character string in the input document so that higher evaluation is provided to a document in which unique character strings with higher weight appear many times. Such a system and method do not require vocabulary information or grammatical information, which run into difficulties when meeting new words or phrases, and allow a document search to be performed against a vague request of a user for document search.

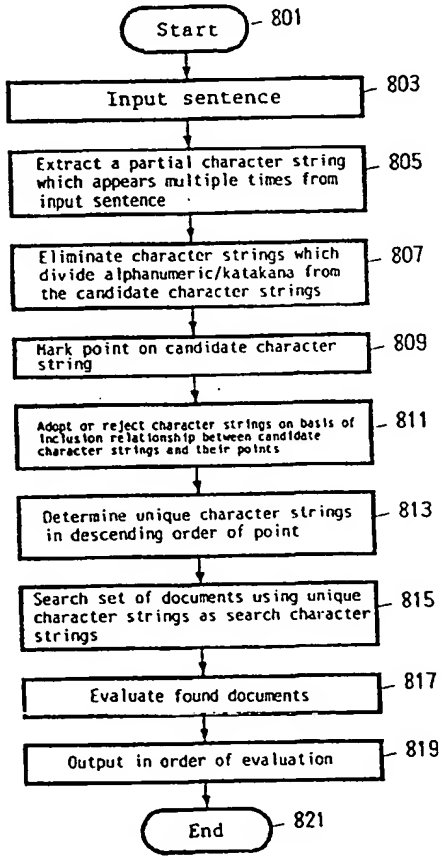

FIG. 10

EP 0 802 492 A1

Description

The present invention relates to a system and method for searching a large volume of documents stored in a computer. More particularly, it relates to a system and method for searching for documents similar to a document at a high speed while allowing desired ambiguity.

In search means for locating a document of interest from a set of electronic document texts, it is widely practiced to input a search expression in which character strings representing contents of interest are connected by logical operators such as AND, OR. or NOT.

[Example]

(computer OR personal computer) AND search

This method places all operations for converting the contents of interest into the search expression into the hands of the user. It is troublesome to think of suitable search character strings, to construct a suitable search expression, and to input the search expression, and the quality of result greatly depends on the skill of the user.

A method is also practiced to weight each search character string, and to sequentially output documents beginning with the ones containing higher weight.

[Example]

Computer, 60 personal computer, 60 search, 100

This method also places all operations for converting the contents of interest into the search expression into the hands of the user. It is troublesome to think of suitable search character strings, to construct a suitable search expression, and to input the search expression. Accordingly, the user must fully understand the contents of documents to be searched, and terms being used. Even if the user vaguely thinks "I want to read sentences with such sense", such request of the user is difficult to be attained.

On the other hand, there is a technology as described in Published Unexamined Patent Application (PUPA) No. 6-124305 to input natural language for search, to extract search keys, and to perform a search based on the extracted search keys. Such a search requires a search key dictionary. In a method of performing extraction based on vocabulary information (word dictionary) such as the search key dictionary or grammatical connection rules, the word dictionary or the grammatical connection rules are stationary so that a new word such as "TOYSARUS" ("TOYSARUS" is a trademark of Toysarus Inc.) or a phrase such as "footprint of dinosaur" is difficult to be extracted as a unique character string. In addition, the concept of "feature" of those contents included in a document changes with the times. For example, while, previously white-collar workers always wore suits when they came to the office, recently, there are many cases where they may not wear suits because many companies adopt a casual day system. To meet such problems, it is necessary to continuously update the word dictionary to include new words, and new trademarks, trade name, and product names, and to keep pace with the change of the times. However, such work requires enormous labor, and the region for storing the dictionary or the like is increased as the update or addition of new words or the like is performed, which in turn affects search speed.

In addition, PUPA 6-223114 describes a method for processing character strings by using frequency of appearance of word, somewhat analogous to the present invention. However, such technology is to determine the type of a document, or to extract keywords for search by searching a registered word list (word dictionary) for whether words in the document exist in it, and is not the technology as in the present invention to investigate frequency of appearance of a word or character string in an input document and a comparison document, and to utilize the frequency of appearance in both of them (in the word list, one word appears only once so that it is meaningless to investigate the frequency of appearance). Thus, a stationary word dictionary becomes necessary so that there still remain the above-mentioned problem that a new word or phrase is difficult to be extracted as a unique character string. In addition, since this technology detect keywords based on stationary word dictionaries by category, if there are multiple documents describing, for example, methods for searching documents, there is a high possibility that all the keywords being extracted are very similar ones such as "search", "character string", and "high speed", and therefore it is difficult to extract keywords for differentiating each document.

Accordingly, the present invention provides a document search method and system as defined by the attached claims.

The approach described herein enables one to build a search system allowing the input of complicated search intention with very simple operations such as clicking of a button, without the need to think of or input a search expression. Such a search method is close to human sensation and allows the easy input of complicated and abstract inten-

tions for search. This reduces the labor of the user for thinking of or inputting a search character string or a search expression, can be used by everybody, and can perform a search even if the user does not accurately understand a keyword to be used for the search.

The search method requires less storage capacity, and extracts a unique character string at a high speed: moreover, the search method relatively and dynamically extracts a unique character string without using vocabulary information or grammar information.

The approach described herein 1) extracts a unique character strina (a character string characterizing the sentence when viewing it from the entire set of documents) from an input sentence: 2) allocates a suitable matching factor to each unique character string for ambiguity search: and 3) evaluates the located documents by using appearance frequency information in the input sentence as weight, and rearranges them in the order of evaluation.

The "Input sentence" may be a collection of meaningful sentences in a language such as Japanese or English, or the entire content of a sentence or one paragraph. In addition, it may be a sentence in which multiple languages such as Japanese and English are intermixed. Furthermore, the "unique character string" may be a character string characterizing the sentence when viewing it from the entire set of documents, or when comparing it with another sentence. When the unique character string is analogized to a plain example, it resembles a word qualifying a person to identify a specific person in a party where a number of unfamiliar persons gather. If most of the persons attending the party wear glasses, then the words "wearing glasses" are not the (distinguishing) feature of that person. On the other hand, if most of persons attending the party are in casual clothes and that person wears a suit, then the words "wearing a suit" are a significant feature of that person.

A unique character string extracted from the input sentence is weighted by the appearance frequency information, and ambiguously searched. When a set of sentences are previously arranged as an index file by extracting appearing position information of N-character chains, high speed search can be performed for every character string in a document. Since extraction of word by morpheme analysis is not performed, no maintenance is necessary, documents can be registered at a high speed, and ambiguity search can be performed for searching character strings with similar arrangement of characters.

Viewed from one aspect, the present invention provides a method for identifying a unique character string contained in an input document which is input into a computer system, said computer system being able to search comparison document stored in a storage medium, the method comprising the steps of: associating and managing position information for a position in the comparison document where a partial comparison document character string extracted from the comparison document exists with the partial comparison document character string; extracting a partial input character string from the input document, and determining them as a candidate character string; identifying a partial comparison document character sting which matches a part of the candidate character string with a predetermined similarity factor or higher: identifying position information associated with the partial comparison document character string which matches with the predetermined similarity factor or higher; and recognizing the candidate character string as the unique character string by comparing appearance frequency information on a part of the candidate character string appearing in the input document with the position information, and evaluating amount of feature of the candidate character string. When such unique character string is recognized, the type of the document can be intuitively understood.

A comparison document stored in the storage medium may represent not only a document stored in a storage device in the computer, but also a document stored in another system but able to be searched by this computer. In addition, the document may be a single document, or multiple documents, or parts of single or multiple documents (a title, a body excluding the title, a footnote or the like). Moreover, in the case of multiple documents, it may be a set of documents including the input document, or a set of document extracted by a search or the like. The contents of document may be of a natural language or a program language.

In addition, an input document may represent not only an entire document in a natural or program language stored in the storage device in the computer, but also a document the whole of which is stored in another system, but parts of which are extracted and input in this computer. In addition, the document may be a single document, or multiple documents, or parts of single or multiple documents. Moreover, the input document may be extracted parts of the comparison document.

The partial input character string being extracted from the input document may represent not only a fixed character string of N characters of a non-delimiter language (N is a natural number of 1 or more) or a variable character string of N or more characters of a non-delimiter language, but also one or more words of a delimiter language. It may be keywords extracted from the input document based on the vocabulary information (word dictionary) or grammatical connection rules.

A part of the candidate character string may denote all of a candidate character string, while frequency information may represent information relating to the number of appearances of a part of the candidate character string in the input document, the comparison document or the like, and may be not only the number of appearances derived by investigating all of the documents, but also information based on the number of appearances in the sample of each document.

In addition, as described with respect to a preferred embodiment of the present invention, it may represent information derived by converting a value relating to the number of appearance in the document, an example of such value being one representing the size of position information data (size of position information file shown in Figure 3 in bytes) corresponding to the N-character chain in the candidate character string quantized to Q level (Q is a constant), or a value which is such value being compressed.

Associating and managing position information for a position in the comparison document with a partial comparison document character string is desirably achieved by a position information file as shown in Figure 3 of the preferred embodiment of the present invention, but may be any other management using a table or information for pointing to an information storage position as long as the position information in the comparison document is associated with the partial comparison document character string.

The amount of feature corresponds to the points or score of the candidate character string as indicated in the preferred embodiment of the present invention. but the concept is not limited to such, and it may be possible, for example, to detect during calculation of point or score of a candidate character string that the candidate character string meets a criterion that it is recognized as denoting the unique character string, and to recognize it as a unique character string.

For evaluating the amount of feature, it is possible to set various conditions such as a case where the upper X character strings with high feature scores (representing a feature of the input document) are evaluated as the unique character string of the input document in question. a case where a candidate character string with an amount of feature exceeding a threshold is evaluated as the unique character string of the input document in question, or a case where a candidate character string including upper X character string and having amount of feature exceeding a threshold is evaluated as the unique character string of the input document in question. The identified unique character string may be used for search as is, or further selected by other conditions (for example, adopted or rejected based on the degree of overlapping).

Viewed from another aspect the present invention provides a method for searching a document to be searched which has character strings similar to a partial input character string existing in an input document input in a computer from a plurality of documents to be searched searchably stored in the computer, the method comprising the steps of: extracting a partial character string from the input document, and determining them as a candidate character string; evaluating amount of feature of the candidate character string through comparison between appearance frequency information on a part of the candidate character string appearing in the input document and appearance frequency information on a part of the candidate character string appearing in the comparison document to recognize the candidate character string as a unique character strings: and searching a document to be searched having character strings similar to the unique character strings from the plurality of document to be searched.

Character strings similar to the unique character string typically represent character strings resembling with a predetermined similarity factor or higher (including a character string with the similarity factor of 100% - ie completely matching). A search as described herein may include not only a search method for ambiguity search as described in the preferred embodiment of the present invention, but also any and all search methods which can search a document from character strings.

Comparison between appearance frequency information as described herein is calculated by a calculation formula based on "appearance frequency information in input document/appearance frequency information in comparison document" as the simplest example, but may be replaced by various calculation formulae (some of which are described in the preferred embodiment of the invention also).

Viewed from another aspect, the present invention provides a method for identifying a unique character string contained in an input document which is input into a computer system, said computer system being able to search a comparison document stored in a storage medium, the method comprising the steps of: extracting a partial input character string from the input document, and determining them as a candidate character string; and evaluating amount of feature of the candidate character string through comparison between appearance frequency information on a part of the candidate character string appearing in the input document and appearance frequency information on a part of the candidate character string appearing in the comparison document to recognize the candidate character string as a unique character strings.

Viewed from another aspect, the present invention provides a method for evaluating similarity between a comparison document and an input document which contains a first unique character string and a second unique character string input in a computer, said computer system being able to search a comparison document stored in a storage medium, the method comprising the steps of: calculating a first weight value corresponding to the first unique character string from appearance frequency information on a part of the first unique character string appearing in the input document: calculating a second weight value corresponding to the second unique character string from appearance frequency information on a part of the second unique character string appearing in the input document; calculating a first appearance frequency value on a part of the first unique character string appearing in the comparison document; calculating a second appearance frequency value on a part of the second unique character string appearing in the

comparison document: and calculating the similarity factor of the comparison document from the first appearance frequency value taking the first weight value into account and the second appearance frequency value taking the second weight value into account.

Viewed from another aspect, the present invention provides a method for evaluating similarity between a comparison document and a unique character string input in a computer system, said computer system being able to search a comparison document stored in a storage medium, the method comprising: means for calculating a weight value corresponding to the unique character string from appearance frequency information on a part of the unique character string appearing in the input document: and means for calculating the similarity factor of the comparison document from the appearance frequency information on a part of the unique character string appearing in the comparison document.

Viewed from another aspect, the present invention provides an apparatus for identifying a unique character string contained in an input document which is input into a computer system, said computer system containing a comparison document searchably stored by the computer, the apparatus comprising: a storage device for storing a position information file which associates and manages position information for a position in the comparison document where a partial comparison document character string extracted from the comparison document exists with the partial comparison document character string: means for extracting a candidate character string from the input document; means for identifying a partial comparison document character string which matches a part of the candidate character string with a predetermined similarity factor or higher: means for identifying position information which is associated to the partial comparison document character string with the predetermined similarity factor or higher in the position information file; and means for recognizing the candidate character string as the unique character string by comparing appearance frequency information on a part of the candidate character string appearing in the input document with the position information, and evaluating amount of feature of the candidate character string.

Viewed from another aspect, the present invention provides an apparatus for searching a document to be searched having a character string similar to a partial input character string which exists in an input document input in a computer from a plurality of documents to be searched which are searchably stored in the computer, the apparatus comprising: an input device for identifying the input document and instructing execution of search; means for detecting from the input device the fact that the input document is identified and that the instruction of search is input; means for extracting a candidate character string from the input document in response to the detection of the fact that the input document is identified and that the instruction of search is input: means for calculating amount of feature through comparison between appearance frequency information on a part of the candidate character string appearing in the input document and appearance frequency information on a part of the candidate character string appearing in the comparison document: means for determining the candidate character string as a unique character string by evaluating the amount of feature: means for searching the document to be searched having a character string similar to the unique character string from a plurality of documents to be searched: and a display device for displaying the document to be searched having a character string similar to the unique character string.

Viewed from another aspect, the present invention provides an apparatus for identifying a unique character string contained in an input document which is input into a computer system, said computer system containing a comparison document searchably stored by the computer, the apparatus comprising: means for extracting a candidate character string from the input document: and means for determining the candidate character string as a unique character string by evaluating the amount of feature of the candidate character string through comparison between appearance frequency information on a part of the candidate character string appearing in the input document and appearance frequency information on a part of the candidate character string appearing in the comparison document.

Viewed from another aspect, the present invention provides an apparatus for evaluating similarity between a comparison document and an input document containing a unique character string input into a computer system, said computer system containing a comparison document searchably stored by the computer, the apparatus comprising: means for calculating a weight value corresponding to the unique character string from appearance frequency information on a part of the unique character string appearing in the input document: and means for calculating the similarity factor of the comparison document from the appearance frequency information on a part of the unique character string appearing in the comparison document and the weight value.

Viewed from another aspect, the present invention provides a storage medium readable by a computer for storing a program which identifies an input document input into a computer system, said computer system containing a comparison document searchably stored by the computer, the program comprising: program code means for directing the computer to extract a partial character string from the input document and making it a candidate string; and program code means for directing the computer to determine the candidate character string as a unique character string by evaluating the amount of feature of the candidate character string through comparison between appearance frequency information on a part of the candidate character string appearing in the input document and appearance frequency information on a part of the candidate character string appearing in the comparison document.

Such a storage medium may comprise a floppy disk, a CD-ROM, an MO, a PD or a storage device connected to

a network. The program code may be divided into a plurality of segments and stored in a plurality of media. In addition, the program may be compressed and stored in a floppy disk. The medium may be loaded on the system through variuous devices such as a floppy disk drive, a modem, or a serial port.

Viewed from another aspect. of the present invention provides a storage medium readable by a computer for storing a program which evaluates similarity between a comparison document and an input document containing a unique character string input into a computer system, said computer system containing a comparison document searchably stored by the computer, the program comprising: program code means for directing the computer to calculate a weight value corresponding to the unique character string from appearance frequency information on a part of the unique character string appearing in the input document: and program code means for directing the computer to calculate the similarity factor of the comparison document from the appearance frequency information on a part of the unique character string appearing in the comparison document and the weight value.

Therefore, as described above, a document search can be performed in response to a vague request of the user for document search. In addition, since a character string search method is provided for extracting a unique character string without using vocabulary information or grammatical information, there is no need for maintenance of the vocabulary information or grammatical information, so that a search system constantly meeting new words or phrases can be provided. Furthermore, since a character string search technique is provided for relatively and dynamically extracting a unique character string, there is provided an advantage of being able to extract unique character strings unique to an input document, rather than to extract similar unique character strings from documents of a special type.

An embodiment of the invention will now be described in detail by way of example only with reference to the following drawings:

Figure 1 is a block diagram showing a hardware configuration:
Figure 2 is a block diagram of processing components:
Figure 3 is a diagram showing the structure of an index file;
Figure 4 is a diagram showing the structure of an index file;
Figure 5 is a flowchart showing an index file creation process;
Figure 6 is a flowchart showing an index file creation process;
Figure 7 is a flowchart of the character search process using the index file:
Figure 8 is a flowchart of the ambiguity search process using the index file;
Figure 9 is a flowchart of the ambiguity search process using the index file:
Figure 10 is a flowchart showing extraction of a unique character string from an input document; and
Figures 11-15 are diagrams showing various screens from the user interface of a preferred embodiment of the present invention.

A. Hardware configuration

Referring to Figure 1, there is shown a schematic view of a system configuration comprising a bus 101 connected to a central processing unit (CPU) 102 having capabilities of arithmetic operation and input/output control, a main memory (RAM) 104 for loading a program and providing an operating environment for the CPU 102, a keyboard 106 for key inputting a command or a character string to be searched, a hard disk 108 for storing an operating system, a database file, a search engine, an index file or the like, a display device 110 for displaying the result of a search from the database, and a pointing device (including a mouse or a track ball) 112 for pointing to any location on the screen and transmitting its position.

Therefore, it will be appreciated that the present invention can be implemented on an ordinary personal computer (PC), a workstation, and the like. In addition, there is shown a storage medium 114 for storing program codes which provides instructions to the CPU or the like in cooperation with an operating system to implement the method described below. The storage medium may be a floppy disk, a CD-ROM, a magnetooptical disk, a PD, and/or a storage device connected to a network. The program codes may be divided into a plurality of segments or compressed, and stored in a plurality of media. The storage medium 114 is loaded on the system through various devices such as a floppy disk drive, a modem, or a serial port.

The operating system is desirably one supporting a GUI multiwindow environment as a standard feature such as Windows (trademark of Microsoft), OS/2 (trademark of IBM) or X-Window system on AIX (trademark of IBM), but may be implemented on a character base environment such as PC-DOS (trademark of IBM) or MS-DOS (registered trademark of Microsoft). It is not limited to a specific operating system environment. In addition, while Figure 1 shows a system in standalone environment. because the database file generally requires a disk device with a large capacity, one may provide as a client/server implementation in which the database file and the search engine are placed on the server machine, and the client machine is connected by a LAN or other suitable network to the server machine, and arranged to have only an input control function for identifying an input document and a display controller for viewing

the result of search.

B. System configuration

Turning now to Figure 2, it should be noted that components represented by respective blocks in Figure 2 are separately or collectively stored as a data file or program file in the hard disk 108 of Figure 1. Those mainly assumed as the database 202 are ones storing a plurality of documents such as a database of newspaper accounts, or a database of patent publications. However, it should be noted the application of the present invention is not limited to a search of a database consisting of a plurality of documents, but includes a search of a single document.

In this case, contents of individual documents are searchably stored, for example, in a text file form. In addition, each document is provided with a unique document number. While document numbers are preferably ascending sequential numbers starting with 1. in the case of a patent publication database, application numbers or laid-open numbers may be used as unique document numbers. In addition, symbols such as "AB" or "&XYZ" may be used to identify individual documents in place of the sequential number. However, since such identification symbol generally requires more bytes than a numeral. it is preferable in practice to identify documents with sequential numbers.

A preferred embodiment of the present can attain high speed search for a document written in either a language which has many kinds of characters, but does not have explicit word delimiters in representation such as Japanese and Chinese (non-delimiter language), or a language which has a relatively small number of characters, and is expressed with explicit delimiters such as English (delimiter language).

In general, since it takes a long processing time to directly search enormous contents such as news articles or patent specifications stored in the database 202, an index file 204 is previously created by an index creation/update module 206 for the contents of all news articles. According to a preferred embodiment of the present invention described later, the thus created index file comprises four files: a character chain file, a position information file, an extended character chain file, and an extended position information file.

The character chain file stores where, in the position information file fixed length chains, variable length chains, delimiter patterns and document numbers corresponding thereto, and position numbers in the document are stored. The extended position information file stores variable length chain numbers and position numbers in the variable length chain. In the preferred embodiment of the present invention, the search can be performed at a high speed by using such a search file. However, since appearance frequency of a character string can be calculated regardless of the format for storing the document, use of such a search file is not necessary in all embodiments of the invention.

In addition, the database 202 may be one storing individual documents as separate files, or a one sequentially arranging all documents in a consecutive single file, provided that individual documents are provided with unique numbers, and the content of individual documents can be accessed with such unique numbers. In the former case, the database 202 manages a table which causes the unique numbers for individual documents to correspond to actual file names storing the documents. In the latter case, the database 202 manages a table which causes the unique numbers for individual documents to correspond to an offset in the single database file and the size of document.

A search engine 208 has capabilities to search the index file 204 with a search character string given by a search character input module 210 as an input, and to return a document number(s) of document containing the input search character string and a position(s) at which the input search character string appears in the document. The search character input module 210 preferably consists of a dialog box in the multiwindow environment, in the input box of which the desired character(s) to be searched is input through the keyboard 106.

Furthermore the search character input module 210 can be used to identify an input document from which a unique character string is extracted. Specifically, titles of input documents are displayed on a display screen, and, when the user selects a title with the pointer of the pointing device 112 such as a mouse, the system recognizes that a document corresponding to the displayed title is selected. The input document may be specified by directly entering information sufficient to identify the document to be input through the keyboard 106.

In addition, the search character string input module 210 can set the amount of feature of a unique character string (extracting a candidate character string as a unique character string when the point of the candidate character string exceeds a preset threshold) as described later, or the number of character strings in a unique character string (extracting candidate character strings as the unique character string from those with higher point by the predetermined number). Furthermore, the search character string input module 210 enables a user to input the similarity factor for ambiguity search in a numerical value of 0 - 1 (which may be a value of 0 - 100 on the basis of percentage). Thus, the search character string input module 210 displays a slider or scroll bar having a handle which indicates any position between 0 and 1. The handle or slider may indicate, for example, 1 as the default, and may be operated to indicate another value by dragging and moving the handle with the mouse 112.

A result display module 212 accesses the database 202 based on the document number as the result of a search given by the search engine 208, and the value of position at which the search character appears in the document, and displays a line corresponding to that position in the document preferably in a separate result display window. If the

search result cannot be contained in one screen of the window, a scroll bar appears so that the user can sequentially view the search result by clicking it.

Furthermore, in the preferred embodiment of the present invention, the result display module 212 has a capability to display the extracted unique character string once on the display 110. The user may add, delete or modify the unique character string, change weighting on each unique character string, or set a condition such as AND or OR on the unique character string, and then may perform search by using the unique character string after modification.

C. Operation

Figure 10 shows the steps of the search method in the preferred embodiment of the present invention. First, the process starts with step 801 where a sentence is first input (step 803). Then, a candidate character string is determined by extracting a partial character string which appears multiple times from this input sentence (step 805). Then, character strings which divide alphanumeric/katakana are eliminated from the candidate character strings (step 807). Subsequently, points (of the amount of feature) are marked on the remaining candidate character strings for indicating how much each of the candidate character string constitutes the feature of the input document (step 809). Then, the character strings are adopted or rejected on the basis of the inclusion relationship between the candidate character strings and their points (step 811). In addition, the unique character strings are determined, for example, in the descending order of points (step 813). Then, the entire set of documents is searched by using the determined unique character strings as the search character strings (step 815). Then, the documents found by the search are evaluated (step 817), and the titles of documents or the like are output in the order of evaluation (step 819).

Prior to the description in detail of each process step, to facilitate understanding by the reader, an example of operations of the user and the system is shown in Figures 11 through 15. This example is to search a database containing a plurality of news articles. Here, it uses a database of news articles of Nihon Keizai Shimbun for one year for which IBM ("IBM" is a trademark of IBM Corporation, U. S. A.) is granted a license for use of copyright from Nihon Keizai Shimbun-sha ("Nihon Keizai Shimbun" is a trademark of Nihon Keizai Shimbun-sha).

(1) The user inputs "Olympics" in an entry 901 shown in Figure 11 through the keyboard 106, and press the Enter key, or clicks the button for Execute Search 931.

(2) The system detects the user input, and searches articles containing a character string "Olympics" by a conventional search process or the ambiguity search process described later.

(3) Then, the system outputs the result of the search on the screen. Specifically, a list of titles 927 on various articles relating to Olympics such as Mathematics Olympics, a store called Olympic, or Nagano Olympics is output into a window 909 in the order of matching factor together with serial numbers 921, matching factor 923 and the date of articles 925. In the embodiment, a document with a high matching factor 100 is selected, and data identifying that document is stored. In addition, the content of a document with the highest matching factor is caused to be displayed in a window 907, and its title or the like is displayed in a window 905. Also, the title or the like of the document currently displayed in the window 907 is highlighted in a window 909.

(4) The user obtains the content of the document in the window 907 by clicking the title in the list of titles in the window 909 or the like. Then, after reading several articles, the user selects an article "Olympics Edition - Successful Olympic Winter Games, Appealing Environment Conscious Nation" from the window 909, and clicks the button. While the user wishes to read articles on an Olympic Winter Games held several years ago, he or she can know that the keyword for articles he or she wishes to read is "Lille Hammer" with this article.

(5) The user clicks Search Similar button 947 to read articles resembling this article (it may be possible to perform the search again for the set of documents extracted in the search for Olympics by entering "Lille Hammer" in the entry 100).

(6) The system detects this input, and performs the search using "Olympics Edition - Successful Olympic Winter Games, Appealing Environment Conscious Nation" as the input. Specifically, a unique character string is extracted from that article, and the similarity search is performed by using that unique character string.

(7) The system displays the result of search on the screen. Specifically, the list of titles is sequentially output in the window 909 from the highest matching factor. While, in the embodiment, the content of the document with the highest matching factor is displayed in the window 907, and its document number, title or the like are displayed in the window 905, it may be possible to display the content of a document with the next highest matching factor in

the window 907. This is because the document with the highest matching factor becomes the input document used for the search. In the window 907, a character string matching or similar to the unique character string in the content of the searched document is highlighted. Displayed in the window 903 is a title of the search for accessing the result of a search. Here, information such as the serial number, the number of documents, and the searched titles is displayed. Here, when the title of "Olympics" previously searched is clicked, information similar to that in Figure 11 is displayed again in the windows 907 and 909.

(8) After the user reads several articles output as the result of search by scrolling the window 909, then he or she wishes to read articles on the snow-board, and selects the article of "Issue on Olympics: Snow-board - Discussion on Adoption as Formal Event ..." in the window 909, and clicks the button.

(9) The user clicks Search Similar button 947 to read articles similar to that article.

(10) The system detects this input, and performs again the search using "Issue on Olympics: Snow-board - Discussion on Adoption as Formal Event ..." as the input.

(11) The system outputs the result of search as shown in Figure 15 on the screen.

The user interface in the preferred embodiment of the present invention has various additional functions. For example, a pull-down menu 911 is provided for inputting a search condition such as AND or OR, or selecting the number of documents to be extracted as the set of a search result, or an allowable similarity factor. A pull-down menu 913 is provided for selecting a matching factor of character string in performing the ambiguity search. In addition, a pull-down menu 915 is provided for selecting whether the subject of the search is the entire set of documents or a partial set of documents such as a set of searched documents. When the search is performed again for the set of searched documents, unique character strings are extracted by comparing the input document and a set of documents as the result of a search limited to a category. Thus, it is possible to extract a character string which is a feature of the input document from a plurality of documents containing similar contents. In addition, the pull-down menu 915 enables a user to select searching for a limited part of a document such as searching for only titles, instead of the entire document. In this case, it can be indicated that it is a character string contained in the title by setting a flag at position information in a position information file, by creating an index file only with titles, by embedding characters or symbols in the document for identifying the title and the body and detecting them to exclude the body from the subject of search, or by causing the title to exist at a fixed area such as the character or line where the title is positioned in the document and performing the search only for that area. Finally, a pull-down menu 917 is provided for selecting whether a search is performed by differentiating the upper and lower cases of the alphabet.

A button 933 is provided to initialize the pull-down menus 911 to 917. For example, when the user has changed the pull-down menu 913 to 80%, and the pull-down menu 915 to the document set 1 (the set of documents as the result of search by the character string "Olympics"), the pull-down menus 913 and 915 are returned to the initial state of 100% and the entire document, respectively, by clicking the initialization button 933. A button 935 is a delete search result button. While the system stores information for identifying a document set as the result of search, clicking of this button causes the system to release the information on the document set highlighted in the window 903 at present, and to delete the titles of the document set from the window 903.

A button 941 is provided to scroll the document so as to display a next unique character string (or matched character string, or similar character string) in the document. A button 943 is provided to display a document with the next higher similarity factor, while a button 945 is to display a document with the next lower similarity factor.

In the sequence of steps described above, (1) through (4) are performed by a known search approach, or an approach for ambiguity search described later. Steps (5) through (11) are described in the following.

D. Extraction of unique character string

In the search method described herein, a unique character string is first extracted from an input document. The unique character string is extracted in accordance with two strategies: (1) extracting a character string containing a character string which has an appearance frequency in the input sentence higher than that in the entire set of documents: (2) extracting a character string which is meaningful even if it is solely extracted.

For Strategy 1, the above-mentioned index file is used. The index file holds all N-character strings in the entire set of documents, and position information data of their appearance in a unique format. The size of position information data changes substantially proportional to the appearance frequency of corresponding N-character strings in the entire set of documents, and can be searched at a high speed in view of the structure of its index. Then, in search, the size of position information data is utilized as a value indicating the appearance frequency of an N-character string in the

entire set of documents.

Detailed steps are described in the following.

D1. Creation of candidate set for unique character string

A candidate set for the unique character string is formed in accordance with the following rules.

[Extraction rule 1] Partial character strings of N characters or more appearing in the input document two or more times are extracted, and added to a candidate set for the unique character string. Parts of symbol characters such as " ", ";", and "(" are excluded from the subject. N is the number of characters in the character chains held in the index file. While N is set to 2 (N = 2) in the preferred embodiment of the present invention, in a delimiter language such as English, each word is extracted as a partial character string. In the extraction, it may be allowed that a plurality of words are converted to one word by meaning these words have such as conversion of "display", "display device", "CRT" or the like to "display device", and then the partial character strings are extracted. In addition, if desired, it may be possible to perform normalization such as conversion from upper case to lower case, conversion from double byte characters to single byte characters, conversion from the plural to the singular, or conversion of tense from the past or past perfect to the present. In addition, it is possible to perform effective extraction of the unique character string at a high speed by excluding character strings such as "a", "the", or "is" which do not constitute the feature of the document based on previous experience.

[Exception rule 2] When the partial character string extracted by the extraction rule 1 divides continuation of an alphanumeric into K characters or less at its start/end position, the partial character string is excluded from the candidate set of unique character string. For example, if K characters are 3, it is possible to prevent "vision" from being extracted from "television".

The purpose of this step is to extract a character string which has a strong relation between characters, that is, a character string which is meaningful even if it is solely extracted as defined by Strategy 2. The exception rule 2 is utilization of experimental knowledge on notation that, when continuation of alphabet is finely divided, it becomes meaningless. In addition, it becomes possible to absorb inflection in English or the like.

D2. Marking point on unique character string candidate

A higher point or score is marked on a candidate for the unique character string (abbreviated to a candidate character string) with less appearance frequency in the entire set of documents, but higher appearance frequency in the input sentence. Thus, the simplest calculation formula is:

[Equation 1]

$$\text{Amount of feature} = \text{Appearance frequency of candidate character string in input sentence} / \text{Appearance frequency of candidate character string in entire set of documents}$$

In addition, when the number of characters in the input sentence and the entire set of documents is taken into account, it can be replaced by the following calculation formula.

$$\text{Amount of feature} = (\text{Appearance frequency of candidate character string in input sentence} * \text{number of characters in entire set of documents}) / (\text{Appearance frequency of candidate character string in entire set of documents} * \text{number of characters in input sentence})$$

Since the position information file shown in Figure 3 is used in the preferred embodiment of the present invention, a point can be marked according to Equation (1) in such a manner that (1) a higher point is marked on a candidate character string containing an N-character chain with less appearance frequency in the entire set of documents, but higher appearance frequency in the input sentence, and (2) a higher point is marked on a candidate character string

with higher appearance frequency in the input sentence.

count i : Appearance frequency of i-th candidate character string in input sentence

Ncount i j : Appearance frequency of j-th N-character chain of i-th candidate character string in input sentence (in the case of a delimiter language such as English, appearance frequency of that word in input sentence)

Nsize i j : Value of size (number of bytes) of position information data corresponding to j-th N-character chain of i-th candidate character string (the number for position number in document of the position information file shown in Figure 3 can be utilized as the appearance frequency information of N-character chain in the entire set of documents) quantized to Q levels (Q being a constant) (in the case of a delimiter language such as English, the position information of that word is used)

Nnum i : Number of N-character chains contained in i-th candidate character string (Number of characters - N + 1)

[Equation 3]

$$\text{Point of i-th candidate character string (amount of feature 1)} = \sum_{J=1}^{Nnum\,i} (Ncount\ i\ j/Nsize\ i\ j)/Nnum\ i \times count\ i \times \underset{i,J=1}{Max}(Nsize\ i\ j)/\underset{i,J}{Max}(Ncount\ i\ j) \quad \ldots (1)$$

This point marking strategy corresponds to Strategy 1. As described above, the size of position information data is substituted for the appearance frequency of the N-character chain in the entire set of documents, and the quantization is to adjust for the difference of granularity between the units for Ncount and Nsize. The purpose of multiplication by Max (Nsize i j) and division by Max (Ncount i) is to adjust for the difference between the total amount of input sentence and the set of documents. For example, a lower point would be given to common character strings such as "the", "a", and "an".

The method for marking point (scoring) on the candidate character string may be variously modifed by those skilled in the art. For example, the counting of appearance may be performed by adding 1.5 to each appearance of a character string at a position in a document with higher importance such as a heading or title in the input document, while 0.5 is added to each appearance of a character string at a position in a document with lower importance such as footnote or quotation.

This equation is described on the following sample of Japanese documents using a database of news articles of Yomiuri Shimbun for which IBM is granted a license for use of copyright from Yomiuri Shimbun-sha ("Yomiuri Shimbun" is a trademark of Yomiuri Shimbun-sha).

In addition, this equation is described on the following sample of English document.

Sample of English document

[Ranking Search and Fuzzy Operation]

Ranking Search returns a list of documents in the order of the score which is level of relevance to specified search condition. The maximum number of the returned documents is specified by the user program. The Ranking Search allows the user to start looking into documents with the most desirable one, which realizes efficient and effective search task. The following three factors are selectable among the factors to decide the score of documents:

a. Frequency of search terms in the document As the search term appears more frequently in the document, the score of the document gets higher.
b. Frequency of search terms in the whole set of documents As the search term appears less frequently in the whole set of documents (all the documents indexed), the search term contributes to the score of the document more.
c. Weight parameter specified explicitly by the user program

As the weight of the search term is larger, the search term contributes to the score of the document more.

The user program can specify which factors to use: one of them, two of them, or all of them. It is allowed to use none of them, and in that case the score is decided by whether the document contains the search term or not. Usually, specifying a and b is recommended.

The way to choose the documents to be scored is selectable from the following two:

- Strict Boolean operation
  Scoring is done for the set of documents as a result of the traditional Boolean operation.
- Fuzzy operation
  Scoring is done for all the documents containing at least one search term. In this case, the operator is said to be Fuzzy operator such as "Fuzzy AND".

Fuzzy Operation

By Fuzzy AND operation, for example, the result of "A AND B AND C" is evaluated higher in the following order:

- The document containing all of the three
- The document containing two of the three
- The document containing one of the three

By Fuzzy NOT operation, for example, the result of "A NOT B" is evaluated higher in the following order:

- The document containing "A" and not containing "B"
- The document containing both "A" and "B"

The traditional strict Boolean operation has an advantage in speed, but it does not allow to evaluate the intermediate status. By using Fuzzy operation, the intermediate status, such as "The document contains not all search terms but almost all" is evaluated, therefore the result is natural to the way of human thinking. Fuzzy operation can be used in Ranking Search only.

In this case, the extracted character strings are "fuzzy", "search", "document", "operation", "score", "term", "contain", "rank", and "evaluate".

Here, if the first candidate character string is "fuzzy", the second candidate character string is "and", and

[Equation 6]

$$\max_{i,j}(Nsize\ i, j) = 390612$$

Nsize 1, 1 = 3028
Nsize 2, 1 = 169568,
then
Ncount 1, 1 = 9
Nnum 1 = 1
count 1 = 9

[Equation 7]

$$\max_{i,j}(Ncount\ i, j) = 59.$$

Thus, the point (score) of "fuzzy" is
score 1 = 9 / 3028 / 1 x 9 x 390612 / 59
= 177.10.

In addition, since

Ncount 2. 1 = 10
Nnum 2 = 1
count 2 = 10,

the point (score) of "and" is
score 2 = 10 / 169568 / 1 x 10 x 390612 / 59
= 3.90.

Thus, it will be understood that this approach can be utilized for a delimiter language such as English.

D3. Adoption or rejection of candidate character string based on degree of overlapping

Candidate character strings satisfying either of the following conditions are excluded from the set of candidates.

[Condition 1] A character at the second character or later is the first character of another candidate character string with higher point (score).

[Condition 2] A character at the second character or earlier from the last is the last character of another candidate character string with higher point.

If the candidate character string to be excluded contains a character string of a length of N or longer not overlapping any other candidate character strings, it is marked with point according to Step 2, and added to the set of candidates for the unique character string. The purpose of this step is to eliminate a candidate character string having a character string with weak relationship before or after it.

D4. Determination of unique character string from candidate character strings

A character string the score of which is within the upper X, and Y or more is determined to be a unique character string. X and Y are constants.

E. Search with unique character string

The entire set of documents is searched for documents containing the unique character string. In the preferred embodiment this search allows for ambiguity as described later to locate those character strings similar to the unique character string. A search match factor for each unique character string (a search parameter indicating how much ambiguity is allowed) is suitably detemined.

F. Output of number of found documents in the order of evaluation

The found documents are evaluated and arranged in the order of evaluation. The similarity factor of a document is evaluated in such a manner that the number of appearances of each unique character string in the input sentence is used as a weight, and higher evaluation is provided for a document in which unique character strings with higher weight appear many times. The simplest calculation formula can be expressed in which the above-mentioned appearance frequency is used as is, and, if the weight of the k-th search character string (unique character string) is weight k, the similarity factor of document score(d) in a document of number d is

[Equation 8]

$$\text{score}(d) = \sum_{k} (\text{weight } k \times \text{appearance frequency of k-th search character string}).$$

In addition, such a similarity factor may be converted by various function equations so that it is dispersed between 1 and 0. Similarly, for the weight value used for this, it may be also possible to use a value converted by a function so that the appearance frequency is dispersed between 0 and 1.

In the preferred embodiment of the present invention, the similarity factor of a document is evaluated by taking the matching factor of the result of the ambiguity search into account, using the number of appearances of each unique character string in the input document, and using Equation (2) so that a document in which many kinds of unique character strings with higher weight appear many times is provided with a higher evaluation.

When the weight of the k-th search character string (unique character string) is represented as weight k, and the matching factor of the first hit (character string similar to the search character string) in a document of the number d as percent (d, k, l).

[Equation 9]

$$score1(d) = \sum_{k} (weight\ k \times \underset{l}{Max}\ (percent\ (d, k, 1)))$$

$$score2(d) = \sum_{k} (weight\ k \times \sum_{l} percent\ (d, k, 1))$$

$$score(d) = \underset{x}{Max}(score2(x)) \times score1(d) + score2(d) \quad \ldots (2)$$

[0114]

Equation (2) can be substituted to the following equation where g (d) is a suitable function including the length of document d, and increasing with the length of document d such as

Length / (length + C) : C being a constant

In addition, T, W, and S are suitable constants larger than 0 but less than 1.

[Equation 10]

$$f(d,k) = \sum percent(d, k, 1)$$

$$t(d,k) = T + (1 - T) * f(d, k) / (f(d, k) + g(d))$$

$$w(d,k) = weight\ k / \sum weight\ k \times (W + (1-W) * t(d, k))$$

$$score(d) = S \times \underset{k}{Min}\ (w(d, k)) + (1-S) \times \sum_{k} w(d, k)/n(d)$$

"f (d, k)" indicates the appearance frequency for which the result of the similarity search is taken into account. In addition, "t (d, k)" is to normalize the "f (d, k)" to 0 - 1 considering of the length of the document, while "w (d, k)" is added with a weight value.

G. Structure of index file and how to create it

In the preferred embodiment, files are created which index all continuations of characters belonging to a character set C (variable length chains), all continuous N characters not belonging to the character set C (fixed length chains), their positions in the document, and division information in document with the document (a document chain file 302, and a position information file 304). Here, the "character set C" means a predetermined set of characters, and preferably alphabet ('A' - 'Z', and 'a' - 'z'). However, it is contemplated that characters used in other languages such as German, French, Italian, or Russian, may impose a condition that the character should be a single byte, or be alphanumeric including double byte characters, or add several symbol characters, special symbols or the like such as "?", "!" or " ' ". Moreover, the "division information in document" means typically a delimiter in a document sentence such as "." or ",", and a delimiter in a document in a broader sense such as "Chapter 1", "Summary", a blank line, or a blank character (s). Then, files are created in response to the variable length chain, which index all continuous N' characters in all variable length chains (extended fixed length chains) and their positions in the variable length chains with the variable length chains (an extended character chain file 306, and an extended position information file 308).

However, the four files of the document chain file 302, the position information file 304, the extended character chain file 306, and the extended position information file 308 are not necessarily physically different files, but should be stored in such a manner that the content controlled by each file can be logically processed.

G1. Normalization of character string

In a preferred embodiment of the present invention, the first processing necessary for creating the index file is to normalize the character string by processing in the following way. When the document to be searched is particularly a Japanese document file, single-byte and double-byte characters may be intermixed. Then, processing is performed to, for example, replace single-byte characters with corresponding double-byte characters (or vice versa), and lowercases with uppercases (or vice versa). The normalization of character string is not an essential component of the method described herein.

The detail of normalization may be changed to normalize single-byte characters to double-byte characters other than alphabet, or to only normalize non-delimiter languages, or to change the search condition to differentiate single-byte characters from double-byte characters, or according to a user specification.

G2. Extraction of fixed length chain information

The next step for creating the index file is for all characters to be searched in the normalized character string and for those not belonging to the character set C, to extract continuous N characters starting from these characters (hereinafter called "fixed length chains"), and to store them in the index file together with the document number and the position number in document, where $N \geq 1$, and $N = 2$ is suitable for Japanese, Chinese and Korean.

It is desirable not to search continuations of a character belonging to the character set C and an adjacent blank character in order to reduce the size of the index file.

G3. Extraction of variable length chain information

The next step to create the index file is to extract continuations of characters in the normalized character string and belonging to the character set C (variable length chains), and to store them together with the document number and the position number in the document in the index file. The character set C may be defined to be other than alphabet. In such case, the character string may contain a plurality of variable length chains in which character strings are continuous. In such case, while the character string may include a plurality of variable length chains, a variable length chain can be extracted by using a blank, line feed ",", ".", ";", "!", or "?" as a delimiter. For example, in the case of "Boys be ambitious." or "Boys (line feed) be ambitious.", three variable length chains of "Boys", "be", and "ambitious" can be extracted. In the preferred embodiment of the present invention, the case of "line feed" following "-" and not including any blank before or after it, continuation of characters before or after it may be determined to be a single variable length chain. Accordingly, even in the case such as "Boys be ambi-(line feed)tious", three variable length chains of "Boys", "be", and "ambitious" are extracted. In addition, it may be possible to perform normalization such as conversion between uppercase and lowercase, conversion between the plural and the singular, and conversion of tense from the past or past perfect tense to the present tense, as desired.

G4. Position information

In the preferred embodiment of the present invention, each individual document is divided into blocks in a manner that is meaningful in the search, and division information is stored in the index file. The document may be divided into blocks by detecting line feeds, a period, punctuation, "Chapter X", or "Section X", detecting a blank line, or detecting the paragraph number in a patent specification, or a certain number of characters may be incorporated into one block. These blocks are assigned a series of numbers or block numbers. According to the preferred embodiment of the present invention, a specially defined delimiter pattern is stored together with the document number of the document and the position information in the document for characters at the boundaries of blocks.

Several different division methods can be obtained by defining several types of delimiter patterns. However, the delmter pattern snoula be defined not to overlap character chains being extracted from a normalized character string. In the embodiment, since one-byte codes are converted into two-byte codes through the normalization, if two bytes are assumed to be one word, when the value of the word is 255 or less, it is not applicable to a normal character code. Then, any word value between 0 and 255 can be individually assigned to several types of delimiter patterns.

The advantages of storing the division information in such a format similar to that of the character chain are as follows:

- Easy to create and update the index. No particular processing is required for the division information; and
- The capacity of the index is not significantly increased.

For example, the increase of the capacity is significantly smaller if compared with a format to append corresponding block numbers to every position number in a document.

The position number in a document is a unique sequential number in the document block assigned to all characters to be searched in the document. Then, the position number in document for the first character in a character chain is determined to be the position number in document for that character chain. If a fixed length chain is less than N at the end of continuation of characters not belonging to the character set C together with subsequent characters, predefined padding characters such as X'00' are padded to make the number of characters N.

G5. Extraction of extended fixed length chain information

The next step to create the index file is, for all characters in all variable length chains, to extract continuous N' characters starting from these characters (hereinafter called "extended fixed length chains"), and to store them in the index file together with the extended character chain number and the position number in the extended character chain, where $N' \geqslant 1$, and $N' = 3$ is suitable if the character set C is alphabet. The search speed can be improved when an extended fixed length chain is extracted after appending a start mark and an end mark before and after a variable length chain, respectively. For example, when "$" and "¥" are used as the start and end marks, respectively, extended fixed chains "$ca", "cat", "at¥", and "t¥" are extracted from a variable length chain "cat". Then, it becomes possible to eliminate mixing of "communication" or the like as noise in determining matching of "$ca" or the like.

G6. Example or position number in document

For example, it is assumed that a document containing a sentence "data base system-123" at the beginning is contained in the database 202 (Figure 2) . If it is determined not to search blank characters adjacent to the characters belonging to the character set C on the assumption that the above character set C is the alphabet, when a position number in the document is appended to each character in this sentence, they become as follows.

[Table 1]

Position number in document for characters	1234	5678	9	10111213141516171819
Normalized character string	data	base	s y s t e m -	1 2 3.
Delimiter method 1				|

Then, it is assumed that the document number for that document is 1, and that the number of characters N for the fixed length chain is 2. Then, individual fixed length chains (length 2), delimiter patterns and document numbers associated thereto, and position numbers in document are as follows.

[Table 2]

Fixed length chain	Document number	Position number in document
·1	1	15
12	1	16
23	1	17
3.	1	18
3	1	19
Delimiter pattern 1	1	19

Individual variable length chains, document numbers associated thereto, and position numbers in the document are as follows.

[Table 3]

Variable length chain	Document number	Position number in document
data	1	1
base	1	5
system	1	9

Then, it is assumed that the numbers appended to the variable length chains are sequentially 1, 2, and 3, and that the number of characters N' of the extended fixed length chain is 3, individual extended fixed length chains (length 3), variable length chain numbers associated thereto, and position numbers in variable length chains are as follows.

Extended fixed length chain	Variable length chain number	Position number in variable length chain
dat	1	1
ata	1	2
ta	1	3
a	1	4
bas	2	1
ase	2	2
se	2	3
e	2	4
sys	3	1
yst	3	2
ste	3	3
tem	3	4
em	3	5
m	3	6

When it is allowed to append a plurality of variable length chain numbers and position numbers in a variable length chain to the extended fixed length chain, the whole capacity can be compressed, and high efficiency can be obtained particularly for a document with many overlapped words. In addition, since overlapped searches can be eliminated by putting such overlapped character strings together, high speed search can be performed.

G7. Role of division information in document

Now, the usefulness of the division information (delimiter) in document for the search is described.

- Search only for specific block

When a document is composed of the title, the abstract, and the body, for example, it would be a common demand to perform a search only for a specific portion such as the title and/or the abstract. Such a search can be attained by storing delimiter patterns and their position information for the end of title and the end of abstract.

- Search for document with strong association between plurality of character strings

It would be a common demand to perform search with awareness or strength of association between a plurality of character strings in the context. For example, it is anticipated that there is higher possibility of stronger association when the character strings are in the same paragraph rather than when they are merely in the same document, and that the association is further stronger if they are in the same sentence. It becomes possible to search a document in which a plurality of character strings are in the same block by storing delimiter patterns and their position information for the end of paragraphs or sentences so that search with awareness of the strength of association can be performed.

G8. Structure of index file

It is necessary to store the character chain, the delimiter pattern, its document number, and the position number in a document in a manner that can be efficiently extracted in searching. Thus, in this embodiment, as shown in Figures 3 and 4, the index file is composed of four files of the character chain file 302 (a file mainly storing the fixed length chain, the variable length chain, and the delimiter pattern), the position information file 304 (a file mainly storing the document number, and the position number in document), the extended character chain file 306 (a file mainly storing the extended fixed length chain), and the extended position information file 308 (a file mainly storing the variable length chain number, and the position number in variable length chain). The character chain file 302 is arranged to store information on where the fixed length chain, the variable length chain, the delimiter pattern, and the document number 312 and the position number in document 314 associated to them are positioned in the position information file 304. The position information file 304 is arranged to store the document number 312 and the position number in document 314. The extended character chain file 306 is arranged to store information on where the extended fixed length chain, and the variable length chain number 316 and the position number in variable length chain 318 associated to it are positioned in the extended position information file 308. The extended position information file 308 is arranged to store the variable length chain number 316 and the position number in the variable length chain 318.

While the embodiment is described for a document in which the delimiter language and the non-delimiter language are intermixed, those skilled in the art would easily understand that the same approach can be applied to a document only of the delimiter language or a document only of the non-delimiter language. In the case of only the non-delimiter language, in general, since there is no need to take the variable length chain into account, the extended character chain file 306 and the extended position information file 308 are not required (however, variable length chains may exist even in a non-delimiter language document if the document consists of keywords extracted from the abstract as in a patent specification.)

In Figure 3, entries in the character chain file 302 are the fixed length chains, the variable length chains and the delimiter patterns in all documents in the database 202. The entries in the character chain file 302 are preferably sorted in ascending order in the order of code values of normalized character chains so as to enable dichotomizing search. "Delimiter pattern 1", "-1", "12" and the like are individual entries in the character chain file 302. Here, "delimiter pattern 1", for example, collectively indicates delimiters of a sentence or phrase such as ",", or "." and is assigned a special two-byte value.

The position information file 304 of Figure 3 stores at least one document number 312 corresponding to individual entries in the character chain file 302, and at least one position number in document 314 associated to each of the individual document numbers.

To cause the entries in the character chain file 302 to correspond to those in the position information file 304, although not shown, the individual entries in the character chain file 302 have an offset from the beginning of the position information file 304 for corresponding entries in the position information file 304, and information on the size of the entries in the position information file 304. That is, in Figure 3, for example, the character chain file 302 seeks the position information file 304 from its beginning from the information on offset stored therein with respect to "delimiter pattern 1", and reads the number of bytes specified in the size information from the sought position, whereby it is enabled to collectively read with respect to "delimiter pattern 1" position number values in document of 16, 19, ... in the document number 1, position number values in document relating to the document number 2 and position number values in document relating to the document number n, if any. In addition, by storing information indicating the range where the fixed length chains, the variable length chains and the delimiter patterns are stored, it is possible to determine to which of the fixed length chain, the variable length chain or the delimiter pattern the information stored in the character chain file 302 belongs.

Generally, the position number values in document relating to the document number i are stored, for example, in a form of (document number i: 4 bytes), (number of position number in document k: 4 bytes), (first position number in document: 4 bytes), ... (k-th position number in document: 4 bytes). Although, in this example, it is arranged to take 4 bytes for storing the absolute position of the document as a field for storing the position number in a document, it is desirable in practice to store offset from one previous position number in document so that the number of bytes is saved to 1 - 3 bytes. It is also desirable to reduce the file capacity by performing compression through coding. It is true to the fields for storing the document number and the position number in the document.

In Figures 3 and 4, entries in the extended character chain file 306 are the extended fixed length chains in all variable length chains in the character chain file 302. The entries in the extended character chain file 306 are preferably sorted in ascending order in the order of code values of normalized character chains so as to enable dichotomizing search. "dat", "ata" and the like are the individual entries of the extended character chain file 306.

The extended position information file 308 of Figure 4 stores at least one variable length chain number corresponding to the individual entries in the extended character chain file 306, and at least one position number in variable length chain associated to each of the individual variable length chain numbers.

G9. Process for creating index file

Now, referring to Figure 5, the process for creating the index file will be described. This process is one which is performed by the index creation/update module 206 of Figure 2 when initially building the database 202, or adding or deleting a document to or from the database 202.

In Figure 5, step 402 performs an operation to ensure a memory region. This is a process to obtain a work area with a predetermined size on the RAM 104 by calling a function of the operating system.

In step 404, one document is read from the database 202 to the memory region preferably obtained in step 402.

In step 406, the above-mentioned normalization is performed for the document read in step 404.

In step 408, fixed length chains, variable length chains, and delimiter patterns are created by scanning the normalized document. Then, the fixed length chains, the variable length chains, delimiter patterns, the document number of the document, as well as the position numbers in document of the fixed length chains, the variable length chains and the delimiter patterns are stored in the memory region obtained in step 402.

In the process in step 408, as the fixed length chains, the variable length chains, the delimiter patterns, the document number and the position numbers in document are being stored in the memory region previously obtained in step 402, the empty space in thus obtained memory region may be exhausted. Then, in step 410, a process is performed for checking whether or not the obtained memory region is full. If so, in step 412, the fixed length chains, the variable length chains, and the delimiter patterns, all of which are stored in the memory region, and the document number of the document, as well as position information in document of the fixed length chains, the variable length chains and the delimiter patterns are sorted based on, for example, character code values of the fixed length chains, the variable length chains and the delimiter patterns, the document number, and the position numbers in document, and written to the disk 108 (Figure 1) as an intermediate file, whereby the memory region in which data written in the intermediate file is stored is released for use in the subsequent process. Then, the process proceeds to step 414.

If it is determined in step 410 that there still remains a space in the memory region, then the process immediately proceeds to step 414.

In step 414, it is determined that there remain documents not yet read in step 404 in the database 202. If so, the process returns to step 404.

If it is determined in step 414 that all documents in the database 202 have been completely read, then the fixed length chains, the variable length chains and delimiter patterns, all of which are not written to the memory region obtained in step 402 and remain, and the document number of the document, as well as the position numbers in document of the fixed length chains, the variable length chains and the delimiter patterns are also sorted based on the character code values of the fixed length chains, the variable length chains and the delimiter patterns, the document number, and the position numbers in document, and written to the disk 108 (Figure 1) as an intermediate file.

Since writing of the intermediate files in steps 412 and 416 causes a plurality of intermediate files to exist on the disk 108, and each of these intermediate files are previously sorted, step 418 performs a process to create the character chain file 302 and the position information file 304 shown in Figure 3 from these intermediate files with a conventional merge/sort technique, and to store them on the disk 108. In addition, since the character chain may repeatedly appear several times in the original intermediate files, a process is performed here to put the entries of the same overlapped character chain into one, and to associate the related document number and the position number in document to it. Thereafter, the intermediate files are no longer necessary and can be deleted.

In step 420, one variable length chain is read from the character chain file 302 into the memory region preferably obtained in step 402. In the preferred embodiment of the present invention, since the storage position of the variable length chain in the character chain file 302 is stored in the character chain file 302 at the time it is created, it is possible to immediately access the top position of the variable length chain in the character chain file 302.

In step 422, the extended fixed length chain is created by scanning the variable length chain. Then, the extended fixed length chain, the variable length chain number of that variable length chain, and the position number in variable length chain of the extended fixed length chain are stored in the memory region obtained in step 402.

In the process in step 422, as the extended fixed length chain, the variable length chain number and the position number in variable length chain are being stored in the memory region previously obtained in step 402, the empty space in thus obtained memory region may be exhausted. Then, in step 424, a process is performed for checking whether or not the obtained memory region is full. If so, in step 426, the extended fixed length chains, the variable length chain number, and the position information in variable length chain, all of which are stored in the memory region, are sorted based on, for example, character code values of the extended fixed length chains, the variable length chain number and the position number in variable length chain, and written to the disk 108 (Figure 1) as an intermediate file, whereby the memory region in which data written in the intermediate file is stored is released for use in the subsequent process. Then, the process proceeds to step 428.

If it is determined in step 424 that there still remains a space in the memory region, then the process immediately proceeds to step 428.

In step 428, it is determined that there remain variable length chains not yet read in step 420 in the character chain file 302. If so, the process returns to step 421.

If it is determined in step 428 that all variable length chains in the character chain file 302 have been completely read, then the extended fixed length chains which are not written to the memory region obtained in step 402 and remain, the variable length chain number, and the position number in variable length chain are also sorted based on the character code values of the extended fixed length chains, the variable length chain number and the position number in variable length chain, and written to the disk 108 (Figure 1) as an intermedate file.

Since writing of the intermediate files in steps 426 and 430 causes a plurality of intermediate files to exist on the disk 108, and each of these intermediate files are previously sorted, step 432 performs a process to create the extended character chain file 306 and the position information file 308 shown in Figure 7 from these intermediate files with a conventional merge/sort technique, and to store them on the disk 108. In addition, since a character chain may repeatedly appear several times in the original intermediate files, a process is performed here to put the entries of the same overlapped character chain into one, and to associate the related variable length chain number and the position number in a variable length chain to it. Thereafter, the intermediate files are no longer necessary and deleted.

H. Search process by using index file

Now, an example of a process for performing a character string search by using the index file created as above will be described by referring to the flowchart of Figure 7. In step 502, first, a process is performed to display, for example, a dialogue box with an input box, and to prompt the user to input a search character string in the input box.

When the user inputs the search character string in the input box, and clicks the OK button, the search character string is normalized, if required, and then, in step 504, a fixed length chain and a variable length chain of N characters are created based on the same rule when the index file is created from that search character string.

In step 506, the fixed length chain is searched from the character chain file.

In step 508, if it is determined that no fixed length chain is found, a message box is preferably displayed in step 526 for indicating that the search character string cannot be found, and the process ends.

In step 508, if it is determined that a fixed length chain is found, since the position information file returns one or more document numbers and at least one position number in document at that document number, this information is stored in step 510 in a predetermined buffer region in the main memory or on the disk for the subsequent process.

In step 512, it is determined whether all fixed length chains created from the search character string have been searched. If so, the process proceeds to step 514. If not, the process returns to step 506 where the search process is performed for the next fixed length chain by using the character cnain file.

In step 514, a variable length chain is searched from the extended character chain file and the extended position information file.

In this case, when variable length chains having excess characters before or after it are eliminated, it is possible to avoid noise such as "cat" -> "communication". Specifically, in case where there are three or more characters before or four or more characters after a matched character string, it may be possible to eliminate these characters, to subtract a predetermined value from the similarity factor for one character existing before or after the matched character string as penalty, or to multiply by a predetermined value (positive value less than one).

In step 516, if it is determined that no variable length chain is found, preferably, step 526 presents a message box indicating that no search character string is found, and ends the process. Although, in the preferred embodiment of the present invention, the message is displayed on the display device 110 of Figure 1, it may be possible to transfer the message to another location through a network.

If it is determined in step 516 that a variable length chain is found, because the extended position information file 308 returns one or more variable length chain numbers 316, the position information file 304 returns in step 518 one or more document numbers 312 for subsequent processing based on this information and at least one position number in document 314 at each of these document numbers which are then stored in a predetermined buffer region in the main memory or on the disk.

In step 520, it is determined whether all variable length chains created from the search character string have been searched. If so, the process proceeds to step 522. If not, the process returns to step 514 where the search is performed with the next variable length chain by using the extended character chain file, and the extended position information file.

In step 522, a check is performed on the position information for the fixed length chains stored in the buffer in step 510 and the position information for the variable length chains stored in the buffer in step 518 to store the document numbers containing the character strings matching the search character string and their position numbers in the buffer region. If it is determined that the search character string is found, the contents of documents in the database 202 are accessed in step 528 from these document numbers and the position numbers in document, and the applicable lines for the document in which the document search character string exists are preferably displayed in the individual windows.

If it is determined in step 524 that the search character string is not found, a message box indicating that the search character string is not found is preferably displayed in step 526, and the process ends.

In order to check that the search character string appears in a specific block in the document (for example, the third block), it is sufficient to count delimiter positions in the document which appear until the position where the search character string appears in the document, to check at which block (x-th block) the search character string is positioned in the document, and to compare it with the specified block number.

I. Ambiguity search process

The process shown in Figure 7 is to perform so-called strict search by using the index file. However, according to the present invention, it is also possible to use a so-called ambiguity search process for character strings including those similar to a specified character string in individual documents in the database at a high speed by using an index file. Specifically, this scheme specifies a character string to be searched, and a search accuracy (larger than zero but equal to one or less) to identify documents including "similar character strings" of which the "similarity factor" with the character string to be searched is higher than the specified search accuracy, and positions in document of the "similar character strings".

I1. Human sensation feeling that character strings are similar

When English, a delimiter language, is considered by taking it as an example, English character strings in which the arrangement of characters has a resemblance, and the meaning of which has a resemblance include:

(1) Different expression
"database", "database", "data-base"

(2) Inflection
"communicate", "communication"

(3) Typographical error
In case of "communication":

"comunication" : missing of character

"communication": reversal of character

"communication": excess character

(4) Hyphenation
"communication", "communi-cation"

(5) Variation of phrase
"new technology", "new CMOS technology"

It is common to them that most characters continuously match, but there is a missing character or an excess character. Similar examples may be found in Japanese, Korean, and Chinese.

I2. Rule for determining similar character strings and similarity factor

First, a description is given for a case of a search character string consisting of only the fixed length chain on a rule for determining similar character strings and a similarity factor. It is a general rule to collect those character strings as similar character strings which have the same sequential relationship with characters in an input character string and at positions close to it to some extent from character strings that continuously match each other with the input character strings over M characters or more, and to determine a similarity factor from the number of matched characters and the number of non-matched characters.

First, terms used in the description are defined.

Matched character string:

A section in which a character string to be searched continuously matches the document text over M characters or more. The longest character string is selected from those starting from a same character.

Example:

Character string to be searched: communication

Document text: ... the communi- ...

If M = 2, "communi" is the matched character string. In this case, because or the longest selection, "com" or "commu" is not referred to as a matched character string. In addition, "t" is also not a matched character string because it is less than two characters.

Valid matched character string:

A matched character string of M characters constitutes a similar character string. A valid matched character string in a search character string is called a valid matched search character string; a valid matched character string in a document is called a valid matched document character string. Since the valid matched search character string and the valid matched document character string match for their contents, they are simply called a valid matched character string unless further distinction is required.

Longest non-matched character string length L:

Non-matched characters to be contained in a similar character string should be continuous L characters. L is a constant of one or more.

Now, description is given on how to select a "similar character string" and how to digitize a "similarity factor".

(1) Determination of first valid matched character string

The first matched character string in the order in a document is to be the first valid matched character string, where it is expressed that

s (D. i) is the start position of the i-th valid matched document character string;

e (D, i) is the end position of the i-th valid matched document character string;

s (C, i) is the start position of the i-th valid matched search character string; and

e (C, i) is the end position of the i-th valid matched search character string.

(2) Determination of next valid matched character string

When the i-th valid matched document character string has been determined, the (i+1)-th valid matched document character string is determined in the following manner.

The first matched character string satisfying the following two conditions a) and b) is made the (i+1)-th valid matched character string.

[Equation 11]

$$a) \quad e(D, i) + 1 \leq s(D, i+1) \leq e(D, i) + L + 1$$

This means that up to L characters are allowed as excess characters which may exist between the i-th valid matched document character string and the (i+1)-th valid matched document character string.

(Refer to Example 3 described later.)

[Equation 12]

b) $s(C, i+1) > e(C, i) - (M-1)$

The process is continued until such valid matched document character string cannot be selected.

(3) Determination of "similar character string" and its "similarity factor" (degree of similarity)

When the valid matched document character string cannot be selected any more, a "similarity factor" is calculated from the following equation by assuming that a "similar character string" is from the first character of the first valid matched character string to the last character of the last valid matched character string.

[Equation 13]

Similarity factor =

minimum (number of characters in a character string to be searched

belonging to a valid matched search character string/number of characters

of the character string to be searched;

number of characters in a "similar character string" belonging to the valid matched document character string/number of characters of the "similar character string")

The similarity factor can be calculated from the number of characters not belonging to a valid matched document character string.

[Equation 14]

Similarity factor = 1 -

maximum (number of characters in a character string to be searched not

belonging to a valid matched search character string/number of characters

of the character string to be searched;

number of characters in a "similar character string" not belonging to the valid matched document character string/ number of characters of the "similar character string")

13. How to count number of characters in "similar character string" belonging to valid matched character string

When two characters correspond to the same character in the character string to be searched, the first character is counted as 1, and the second one is counted as 0.5. Otherwise, one character is counted as 1. (Refer to Example 4 described later.)

14. Order of determination of "similar character string"

The first "similar character string" is determined by starting the comparison from the top of a document. When the i-th "similar character string" has been determined, the (i+1)-th "similar character string" is determined by starting comparison from the first character which is behind a character at the top of the i-th "similar character string", and does not belong to a valid matched character string constituting the i-th "similar character string".

A "similarity factor" considerably agreeing with the general human determination can be calculated on whether or not the arrangement of characters resembles one another by setting the constants L and M to suitable values.

When the "similarity factor" attains the maximum value of 1, character strings completely match, i.e., when the

character strings completely match, the "similarity factor" always becomes 1.

15. Process flowchart for ambiguity search

The above process is represented as a flowchart shown in Figure 8. Referring to Figure 8, first, in step 602, input of a search character string is prompted. Also, in step 604, input of similarity factors of 0 - 1 is prompted. Usually, input of the character string and the value in steps 602 and 604 is performed by using an input box and a scroll bar of a single dialogue box.

In step 606, the number i for a valid matched character string is set to 1. In step 608, the valid matched character string is searched. Now, if there is a condition that the length of valid matched character string is M cnaracters or more, it is advantageous that an index file of M-character chain is created in the process of Figure 7. This is because, if such index file already exists, the search for any M-character chain can be performed at a high speed by the dichotomizing the search for the index file. Subsequently, the search for an M-character chain is performed in the index file by shifting the start position for taking the M-character chain in the index character string by one. Then, if the resulting document number is the same as one previous search for the M-character chain, and the position number in the document is sequential, a valid matched character string with a length of M+1 would be obtained. Thus, whenever the conditions that the document number is the same as one previous search for the M-character chain and that the position number in the document is sequential are satisfied, the length of the valid matched character string is incremented by one. However, if nothing is found in the search for the M-character chain using the index file, or if the document number being returned does not match, or if the position number in the document becomes non-sequential, the end position of the valid matched character string would be found.

Sometimes, no valid matched character string is found. In such case, depending on the decision in step 610, the process proceeds to step 626, where it indicates that nothing is found and ends.

When it is determined in step 610 that a valid matched character string is found, the process proceeds to step 612, and s (D, i) to e (D, i) in the document and s (C, i) to e (C, i) in the search character string are marked as the valid matched character string.

In step 614, the (i+1)-th valid matched character string satisfying the conditions

[Equation 15]

$$a) \quad e(D, i) + 1 \leq s(D, i+1) \leq e(D, i) + L + 1, \text{ and}$$

$$b) \quad s(C, i) > e(C, i) - (M-1)$$

is searched by also using the index file. If found, the process returns to step 612 where, for the (i+1)-th valid matched character string, s (D, i+1) to e (D, i+1) in the document and s (C, i+1) to e (C, i+1) in the search character string are marked as a valid character string (increment of i in step 618 indicated to pay attention on the next valid matched character string).

On the other hand, if a valid matched character string is not found in step 616 any more, a similarity factor is calculated in step 620. It is given as explained above by, for example,

[Equation 16]

Similarity factor =

minimum (number of characters in a character string to be searched

belonging to a valid matched character string/number of characters of the

character string to be searched;

number of characters in a "similar character string" belonging to the valid matched character string/number of characters of the "similar character string")

In this case, the "similar character string" is a character string from the start position of the first valid matched character string in the document to the last position of the last valid matched character string.

In step 622, results are selected from the similarity factor calculated in step 620 and that input in step 604. Only

results with the similarity factor equal to or higher than that input in step 604 are displayed in step 624.

In step 624, a process is performed to access contents of documents stored in the database based on the document numbers and the position numbers in a document returned as the result of searches of the index file in steps 608 and 614, and to display lines containing applicable sections.

Although the "similar character string" for one search character string may be simultaneously found in a plurality of documents, it may be found at a plurality of sections even in a single document. Accordingly, it should be noted that steps 606 - 622 are applied to each of such plurality of "similar character strings", and, in step 624, only those of the plurality of "similar character strings" satisfying the conditions for a similarity factor are selected and displayed.

16. Examples of determination on "similar character string" and similarity factor

Examples are given with M = 2, and L = 3.

[Example 1]

```
                                              123456
Character string to be searched C:   ABCDEF

                        123456/8 ...
Document D:   AB.CD.EF ...
```

Since the longest character string first matched is "AB",
the first valid matched character string is "AB"

s (C, 1) = 1 e (C, 1) = 2

s (D, 1) = 1 e (D, 1) = 2.

Since e (C, 1) - (M-1) = 1, the second valid matched character string is searched by comparing a character string starting at or after the second character in the character string to be searched with a character string starting at the third, fourth, fifth or sixth character in the document (because e (D, 1) + 1 = 3, and e (D, 1) + L + 1 = 6).

Second valid matched character string "CD"

s (C, 2) = 3 e (C, 2) = 4

s (D, 2) = 4 e (D, 2) = 5

Since e (C, 1) - (M-1) = 3, the third valid matched character string is searched by comparing a character string starting at and after the fourth character in the character string to be searched with a character string starting at the fifth, sixth, seventh or eighth character in the document (because e (D, 2) + 1 = 5, and e (D, 2) + L + 1 = 8).

Third valid matched character string "EF"

s (C, 3) = 5 e (C, 3) = 6

s (D, 3) = 7 e (D, 3) = 8

Since the end of the character string to be searched is reached, the third is the last valid matched character string.

[Table 5]

AB	CD	EF
1	2	3

AB·CD·EF

1 2· 3

Numerals are the number of valid matched character string.

Therefore, the "similar character string" is "AB.CD.EF" from s (D, 1) to e (D, 3).

"Similarity factor" = minimum (6/6, 6/8) = 6/8 = 0.75

[Example 3]

[Table 8]

	1234
Character string to be searched C:	ABCD

[Table 9]

	1 2 3 4 5 6 7 8 9 101112131415
Document D:	A B X X X X C D X X X X X X .

Since the longest character string first matched is "AB",

[Equation 17]

First valid matched character string is "AB" s (C, 1) = 1 e (C, 1) = 2
s (D, 1) = 1 e (D, 1) = 2

The second valid matched character string is searched by comparing a character string starting at the third, fourth, fifth or sixth character in the document (because e (D, 1) +1 = 3, and e (D, 1) + L + 1 = 6) with a character string starting at or after the second character in the character string to be searched (because e (C, 1) - (M-1) = 1).

Since the second valid matched character string is not found, and the end of the character string to be searched is reached, the valid matched character string is only the first one.

[Table 10]

ABCD
1
ABXXXXCDXXXXXX.
1

Therefore, the first "similar character string" is "AB" from s (D, 1) to e (D, 1).
"Similarity factor" = minimum (2/4, 2/2) = 2/4 = 0.5

The first non-valid matched character after "A" is "X". When the second "similar character string" after "X" is searched:

[Table 11]

ABCD
1
ABXXXXCDXXXXXX.
1

However, since "AB" and "CD" are separated by four characters in the document, and L = 3 in this example, the above "CD" is not considered as a valid matched character string.

[Example 4]

[Table 12]

```
                                        123
Character string to be searched C:      ABC

                    123456789
Document D:         XABBCXXXX
```

Since the longest character string first matched is "AB":

[Equation 18]

First valid matched character string is "AB" s (C, 1) = 1 e (C, 1) = 2
s (D, 1) = 2 e (D, 1) = 3.

The second valid matched character string is searched by comparing a character string starting at or after the second character in the character string to be searched (because e (C, 1) - (M-1) = 1) with a character string starting at the fourth, fifth, sixth or seventh character in the document (because e (D, 1) + 1 = 4, and e (D, 1) + L + 1 = 7).

Second valid matched character string is "BC"	s(C, 2) = 2 s(D, 2) = 4	e(C, 2) = 3 e(D, 2) = 5

Since the end of the character string to be searched is reached, there are two valid matched character strings.

[Table 13]

```
ABC
1
 2__

XAB   BCXXXX
 1    2
1,1, 0.5,1->3.5
```

The "similar character string" is "ABBC" from s (D, 1) to e (D, 2). "Similarity factor" = minimum (3/3, 3.5/4) = 3.5/4 = 0.875

I8. Search of variable length chain

The method for searching a similar character string is described in the above for a case of a search character string consisting of only fixed length chains. When this is extended to a search character string containing variable length chains, it becomes as follows. Here, a variable length chain taken out from the search character string is called

an extended search character string.

First, the following extended character chain is obtained by searching for an extended character chain file, and an extended position information file. The searching method is the same as the method for searching a search character string consisting of only a fixed length chain from a character chain file and a position information file. M' is used as a constant corresponding to the constant M.

(1) Searching a variable length chain matching an extended search character string with a specified search matching factor or higher. In this case, eliminating variable length chains not matching the first character in the extended search character string is effective to reduce noise. In this case, in creating an extended fixed chain, high speed processing can be performed by using "$" indicative of start, creating an extended fixed chain of "$co", and eliminating variable length chains not matching it. However, even when "$" indicative of start is not used, it is possible to identify a start position of a variable length chain from a position number in a variable length chain or information on a delimiter in the extended position information file.

[Example]

Extended search character string: "communication"

Found variable length chain: "commuication"

(2) Searching a variable length chain matching an extended search character string which is newly created by joining extended search character strings with a specified search matching factor or higher

[Example]

Search character string: "data-base"
Extended search character string: "data", "base"
Extended search character string created by joint: "database"

It may be possible to set the number of joints up to two or three. This process enables it to locate the joined "database" from character strings in a document even in a case of a divided search character string such as "data base".

(3) Searching variable length chains satisfying all the following conditions from variable length chains matching extended search character string with matching factor larger than 0

- The first character in variable length chain is included in the matched section.

[Table 17]

[Example] (underline under matched section)

Extended search character string		Variable length chain
"database"	-> o	"data"
"database"	-> ×	"update"
"database"	-> o	"base"

- The first or last character in extended search character string is included in the matched section.

In this case, the process can be performed at a high speed by creating an extended fixed chain of "$co" and "se¥" through use of "$" indicative of start and "¥" indicative of end in creating the extended fixed chain, and by eliminating variable length chains not matching either of them.

[Table 18]

[Example] (underline under matched section)		
Extended search character string		Variable length chain
"database"	-> o	"data"
"database"	-> o	"base"
"database"	-> x	"tab"

[0257]

This process enables one to locate the divided "data" and "base" from character strings in a document even in a case of a joined search character string such as "database". When a variable length chain contains non-matched characters in the number equal to or more than a predetermined number of characters such as when the first character in the extended search character string is included in the matched section (the variable length chain "data" for the extended search character string "database" being applicable to this), or when the last character in the extended search character string is included in the matched section (the variable length chain "base" for the extended search character string "database" being applicable to this), such variable length chain may be excluded from the subject for search. This enables one to increase the search speed by narrowing down the subjects for search.

In the preferred embodiment of the present invention, the variable length chain subject for the processes (1), (2) and (3) described above (that is, (1)+(2)+(3)) becomes the "variable length chain satisfying the conditions" in step 708. However, the conditions may be variously changed to be (1) only, (3) only, (1)+(2) or the like through appropriate setting instead of that used for above processes (1)+(2)+(3).

In determining the matching factor for searching an extended index by looking for the variable length chains of (1), (2) and (3), if an evaluation lower than normal matching but higher than normal non-match is given to reversal of characters, it is effective to search a word in which the position of characters is reversed, which is often found in a typographical error of an English word.

[Table 19]

[Example of reversed characters]

"commu nication" <-> "commn uication"

In this example, a non-matched character string in the search character string is "un", while a non-matched character string in the document character string is "nu". In such case, such non-match caused by a typographical error can be detected by determining, for example, whether or not "u" and "n" of the non-matched character string in the search character string are contained in the non-matched character string in the document character string.

The search method for a search character string containing such a variable length chain is performed in the procedure shown in Figure 9. This procedure is described with reference to an embodiment. In this embodiment, search characters are "data communication", and "data ..... communication", "data comminucation", and "daily communication" exist in a document character string. It is assumed that "data" and "communication" in "data communication" are sufficiently separated.

Information on a character chain file and that on a position information file are assumed to be as follows (M' = 3):

Character chain file	Position information file
1. data	1-1, 2-1
2. daily	3-1
3. comminucation	2-6
4. communication	1-35, 3-7

Extended character chain file	Extended position information file
$da	1-1, 2-1
dat	1-2
ta¥	1-3
a¥	1-4
dai	2-2
ail	2-3
ily	2-4
ly¥	2-5
y¥	2-6
$co	3-1, 4-1
com	3-2, 4-2
omm	3-3, 4-3
mmi	3-4
min	3-5
inu	3-6
nuc	3-7
uca	3-8
cat	3-9, 4-9
ati	3-10, 4-10
tio	3-11, 4-11
ion	3-12, 4-12
on¥	3-13, 4-13
n¥	3-14, 4-14
mmu	4-4
mun	4-5
uni	4-6
nic	4-7
ica	4-8

Here, for the purpose of easy understanding, an unsorted file is shown. Here, in "1. data 1-1, 2-1", "1". indicates the variable length chain number, "data" indicates a variable length chain, and "1-1" and "2-1" indicate "document number - position number in document". In addition, in "$da 1-1, 2-1", "$da" indicates an extended character chain, and "1-1" and "2-1" indicate "variable length chain number - position in variable length chain." Accordingly, "$da 1-1, 2-1" represents the first character in variable length chain number 1 (data) and the first character in variable length chain number 2 (daily).

When the procedure of Figure 9 is started, a search character string "data communication" is input (step 702). Then, a similarity factor is input (step 704). It may be possible to set this similarity factor as default, and to omit its input. Here, a similarity factor 0.80 is input.

Then, a fixed length chain and a variable length search character string are created from the search character string (step 706) . In this example, since the description is made on a document consisting of only delimiter language, there is no fixed length chain.

Variable length chains satisfying the conditions are searched from the extended cnaracter cnain rile and one extended position information file (step 708) . Here, a process is performed for searching variable length chains of above (1), (2) and (3). That is, (1) search is performed for a variable length chain matching an extended search character string with a specified search matching factor or higher. While the search matching factor in this case may be set to the same value as the similarity factor for the entire character string, it is preferable to be lower than the similarity factor for the entire document. For example, since "communication" and "comminucation" match for 10 characters in 13 characters, and three characters do not match, the similarity factor is 10/13 = 0.77 in a simple calculation method of similarity factor (here, for easy understanding by the reader, description is made by using the simple calculation method of similarity factor). The similarity factor between "data communication" and "data comminucation" is 15/18 = 0.83 in the simple calculation method of similarity because they match for 15 characters in 18 characters including delimiter, and three characters do not match. Thus, there are many cases where the similarity factor for an entire character string

becomes higher even if the similarity factor between the variable length chains is low, and this tendency becomes more significant when the character string becomes longer.

In this embodiment, the "specified search matching factor" is set to 0.60 for "data" and 0.72 for "communication". The "specified search matching factor" may be changed depending on the ratio of the number of characters in the search character string to that in the variable length chain.

This is because, as shown in this example, while the similarity factor for entire character string does not become 0.80 or higher unless the variable length chain matches "communication" for 10 characters or more, there is a possibility that, for "data", the similarity factor for an entire character string becomes 0.80 or higher even if only one character matches. However, if too low a matching factor is allowed for "data", the number of matched variable length chains increases and the search speed is affected so that 0.6 is set as the lower limit. In addition, in the preferred embodiment of the present invention, variable length chains not seriously affecting the similarity factor for an entire document are excluded from the subject for variable length chain searching of (1). This enables it to improve the search speed.

In the embodiment, high speed search is made possible by excluding variable length chains which have a number of characters less than the similarity factor x number of characters of variable length chain to be searched (= 0.72 x 13 = 9.36) from the subject for search. This is attained by controlling the number of characters of the variable length chain, for example, as follows:

Extended character chain file	Extended position information file
$da	1-1-4, 2-1-5.

"1-1-4" and "2-1-5" in "$da 1-1-4, 2-1-5" indicate "variable length chain number - position in variable length chain - number of characters in variable length chain".

As variable length chains having the search matching factor of 0.60 or higher for "data" and 0.72 or higher for "communication", "data" (100%) can be found for "data", and "comminucation" (77%) and "communication" (100%) can be found for "communication" by a method similar to that for the fixed length chain.

In the case of a search character string "communication", for example, chains matching a corresponding extended search character string are the following in the extended character chains:

Extended character chain file	Extended position information file
$co	3-1, 4-1
com	3-2, 4-2
omm	3-3, 4-3
mmi	3-4
min	3-5
inu	3-6
nuc	3-7
uca	3-8
cat	3-9, 4-9
ati	3-10, 4-10
tio	3-11, 4-11
ion	3-12, 4-12
on¥	3-13, 4-13
n¥	3-14, 4-14

Then, "3. comminucation" and "4. communication" in the character chain file can be found from the information on the extended position information file.

Then, (2) a search is performed for a variable length chain matching an extended search character string which is newly created by joining extended search character strings with a specified search matching factor or higher. Therefore, variable length chains matching the search character string "datacommunication" with a specified search matching raccor or nigner are searched. In tnis case. wnile tne "specified search matching factor" may be set to the same value as the similarity factor the entire character string, it may be lower than the similarity factor for the entire document. For example, although, when the search character string is "data base system", there exist three extended search character strings created by joining of "database", "basesystem" and "databasesystem", the impact of these joined search character strings on the similarity factor for an entire document varies depending on the number of characters in the joined search character strings. In the embodiment, because the "specified search matching factor" of (2) is set to 0.80, there is no variable length chain matching the search character string "datacommunication" with the specified search matching factor or higher.

Then, variable length chains satisfying all conditions of "1. the first character of variable length chain is contained in the matched section", and "2. the first or last character of extended search character string is contained in the matched section" are searched in (3) variable length chains matching the extended search character string with the matching factor of 0 or higher. The variable length chains meeting these conditions are "data", "comminucation" and "communication" as in (1).

Referring to Figure 9 again, in step 710, it is determined whether or not a variable length chain is found. In the embodiment, "1. data", "3. comminucation" and "4. communication" have been found. Numbers of these variable length chains are stored in a buffer (step 712). In the embodiment, the variable length chain numbers 1, 3, and 4 are stored. In the preferred embodiment of the present invention, the variable length chain character strings "data" and "communication" of the search character string "data communication" are assigned a number (variable length chain search character string number), respectively. The variable length chain numbers 1, 3 and 4 are stored in connection with such numbers. Therefore, the information being stored is (1-1) and (2-3, 4). This (2-3, 4) indicates the variable length chain numbers 3 and 4 of the variable length chains in the document matching the variable length chain search character string number 2 with a certain matching factor or higher.

Then, in step 714, the similarity factor for the entire character string is calculated from the positional relationship between the position information of fixed length chain and the position information of the variable length chain. Specifically, as described above, the variable length chain numbers 1, 3 and 4 in the document are stored for the variable length chain search character string number. Therefore, the variable length chains in a document which may be joined are 1-3 and 1-4 ("1." corresponds to "data", "3." to "comminucation", and "4." to "communication").

Since, when the contents of the character chain file 302 and the position information file 304 are referenced,

Character chain file	Position information file
1. data	1-1, 2-1
3. comminucation	2-6
4. communication	1-35, 3-7,

combinations of (1-1, 2-1) - (2-6), and (1-1, 2-1) - (1-35, 3-7), that is, combinations of

(1-1) - (2-6),
(2-1) - (2-6),
(1-1) - (1-35),
(1-1) - (3-7),
(2-1) - (1-35), and
(2-1) - (3-7)

become candidates.

However, cases where 1. document numbers are different, where 2. condition L = 3 is not satisfied, and where 3. position numbers in a document are reversed, are excluded from the candidates for the calculation of similarity factor. The combination of variable length chains with such conditions is only (2-1) - (2-6). Therefore, "data comminucation" is calculated for the similarity factor. However, the conditions where "2. condition L = 3 is not satisfied", and where "3. position numbers in document are reversed" are employed for a case where order is important as in "data comminucation", but are not employed for a case where order is not important as in searching for character strings which are extractions of keywords (variable length chains) in the abstract from a patent specification.

In the above description, in step 712, the variable length chain numbers are stored in connection with the variable

length chain search character string numbers. However, storing the variable length chain search character string numbers is not required. Combinations of variable length chains can be determined without information on the variable length chain search character string numbers.

This is specifically described. The contents of the character chain file 302 and the position information file 304 are referenced from the stored variable length chain numbers in step 712.

Character chain file	Position information file
1. data	1-1, 2-1
3. comminucation	2-6
4. communication	1-35, 3-7

The character chain file is divided according to the contents of the position information file.

Character chain file	Position information file
1. data	1-1
1. data	2-1
3. comminucation	2-6
4. communication	1-35
4. communication	3-7

Then, the character chain file is sorted according to the contents of the position information file.

Character chain file	Position information file
1. data	1-1
4. communication	1-35
1. data	2-1
3. comminucation	2-6
4. communication	3-7

If L = 3, it is found that, in view of the contents of the position information file,

1. data	1-1,
4. communication	1-35, and
4. communication	3-7

have no other variable length chain to be combined.

On the other hand, because

1. data	2-1, and
3. comminucation	2-6

satisfy the condition L = 3, they are combined and determined for the similarity factor.

Therefore, the candidates for the calculation of similarity factor are:

1. data	1-1,
4. communication	1-35,
4. communication	3-7, and
1-3. data comminucation	2-1

To prevent duplicated calculation of similarity factor, they are arranged by the variable length chain number.

1. data	1-1

(continued)

4. communication	1-35, 3-7
1-3. data comminucation	2-1

Then, character strings less than search character string (18: including delimiter) x similarity factor (0.80) (character strings less than 14.4) are excluded from the subject for calculation of similarity factor (in practice. it is desirable to perform this before they are arranged by the variable length chain number). Accordingly, the candidates for the calculation of similarity factor become only

1-3. data comminucation 2-1.

Since the search character string "data communication" matches "data comminucation" for 15 characters of 18 characters, and three characters do not match, the similarity factor between them is calculated as 0.83.

The calculation of similarity factor for character string in step 714 can be reduced by storing the similarity factor of variable length chain together with the variable length chain numbers in step 712. That is, (1-1. 00), (3-0. 77) and (4-1. 00) are stored in step 712 and utilized.

In the simple calculation method of similarity factor, the similarity factor may be calculated from the following equation.

[Equation 19]

Similarity factor = (number of characters in variable length chain 1 x

matching factor of variable length chain 1 + number of characters in

variable length chain 2 x matching factor of variable length chain 2 +

... + number of characters of delimiter) / (number of characters in

search character string)

Accordingly, the similarity factor of the embodiment is (4 x 1.00 + 13 x 0.77 + 1) / 18 = 0.83. Whether or not the delimiter is counted as one character may be changed by the design.

Then, in step 716, the input similarity factor is compared to the calculated similarity factor. If there exists no character string with similarity factor higher than the input similarity factor, a display indicating tnat none is found is displayed on tne display 110 (step 720). If it is found, applicable line(s) in applicable document(s) is displayed (step 718). However, the display indicating that none is found and the display of applicable line(s) in applicable documents are not an essential component: rather the information may be transmitted to another computer (including a client). In addition, the display of applicable line(s) in applicable document(s) may display all character strings with the input similarity factor or higher together with their similarity factor, document numbers, position numbers in document and the like, or simply display only predetermined numerals. The order of display may be the sequence of appearance in the document(s) or in the descending order of similarity factor. Moreover, in the case of multiple documents, it may be possible to display predetermined numerals for character strings satisfying conditions in each document. They can be set variously in the design stage.

While a technique has been described for use in the ambiguity search for a document, the same approach may be applied to spelling checking of words in a document. In this case, first, a word in the document not existing in a dictionary is detected by the conventional approach. Then, the detected word not existing in the dictionary is used as a search character string, and the ambiguity search is performed for words existing in the dictionary. Then, words with a certain similarity factor or higher in the ambiguity search are displayed as candidates with correct spelling for the word not existing in the dictionary.

The search for character strings consisting of only variable length chains has been described in the above. For a document in which variable length chains and fixed length chains are intermixed, the similarity factor for the entire character string is calculated in step 714 from the positional relationship between the position information for a fixed length chain and the position information for a variable length chain. This process is described by exemplifying an embodiment. In the embodiment, there exists a search character string "ASEAN123" and "ASEA012" exists in the

document. "ASEAN" and "ASEA" are variable length chains.

In such case, the contents of the character chain file 302 and the position information file 304 are:

Character chain file	Position information file
1. ASEA	1-1
2. 01	1-5
3. 12	1-6

A variable length chain document character string "ASEA" similar to the variable length chain search character string "ASEAN" has been found by the above-mentioned method. For this character string, detection of a valid matched character string and calculation of similarity factor are performed in a similar manner to the method described for fixed length chain.

[Example]

[Table 20]

		12345678
Character string to be searched	C:	ASEAN123
		1234567
Document	D:	ASEA012

[Table 21]

ASEAN123
1 2

ASEA012
1 2

Similar character string = "ASEA012"

Similarity factor = minimum (6/8, 6/7) = 0.75

As described for the calculation of similarity factor for a character string containing only variable length chains, in the calculation of similarity factor, the result of a calculation of a similarity factor for a variable length chain may be used for the calculation of a similarity factor for an entire character string. In the simple calculation of similarity factor, the similarity factor becomes possible to be calculated from the following equation.

[Equation 20]

$$\text{Similarity factor} = (\text{number of characters in variable length chain 1} \times \text{matching factor of variable length chain 1} + \text{number of characters in valid matched character string of fixed length chain} + \text{number of characters of delimiter}) / \text{number of characters in search character string}$$

Accordingly, the similarity factor of the search character string in the embodiment is $(5 \times 0.80 + 2 + 0) / 8 = 0.75$.

and the similarity factor of the document character string is (4 x 0.80 + 2 + 0) / 7 = 0.74. Thus, the similarity factor of the search character string is:

Similarity factor = minimum (0.75, 0.74) = 0.74

In this calculation of similarity factor, the calculation may be performed by changing weight for the variable length chain and the fixed length chain. For example,

[Equation 21]

$$\text{Similarity factor} = (\text{number of characters in variable length chain 1} \times \text{matching factor of variable length chain 1} \times 0.5 + \text{number of characters in valid matched character string of fixed length chain} + \text{number of characters of delimiter} \times 0.2) / (\text{number of characters in search character string} \times 0.5 + \text{number of characters in fixed length chain in search character string} + \text{number of characters of delimiter in search character string} \times 0.2)$$

E9. Application to search equation

The above-mentioned search is an example of a fixed character string. A case where it is applied to a search equation is described. For example, in a search equation such as

(computer OR system) AND communication

(a search equation searching for a document containing "computer" or "system", and containing "communication"), it is conceived to perform the ambiguity search by specifying a search matching factor for each search character string. When the search is performed for every search character string with the matching factor of 80% or higher, for example, the following documents may be found:

a document containing "conputer" and "comnication", and

a document containing "sys-tem" and "communication"

In addition, when the found documents are arranged in descending order of possibility close to the one to be located, it is possible to use the matching factor obtained as the result of search as a cue.

E10. Relationship between structure of index and search for "similar character string"

Ambiguity search for a "similar character string" can be attained at a considerably high speed with the structure of the index described herein by suitably determining the value of M.

Determination of constants N and M

[Table 22]

N	Number of characters in character chain to be stored in index
M	Shortest length of valid matched character string in ambiguity search
L	Longest length of non-valid matched character string in "similar character string" in ambiguity search

Although, if the values of N and N' are increased, the number of types of character chains increases, the volume of data decreases per one character chain, and the search can be performed at a higher speed, and the capacity of index file increases. Sufficient search speed is obtained at N = 2 and N' = 3 for average documents in Japanese, Chinese, Korean, and English.

In addition, if M and M' are determined to be M ≥ N and M' ≥ N', sufficient search speed is obtained in the ambiguity search. In view of the fact that the smaller M and M' are, the finer search can be attained, it is believed to be desirable to set M = N and M' = N'.

E11. Second embodiment for determining similarity factor

[0305]

The ambiguity search of the second embodiment is particularly considered for equilibrium between "the more number of non-matched characters is inserted, the less similarity one feels" and "too much non-matched characters are inserted, then it cannot be felt to be one character string". When a character string matching an input character string, a non-matched character string and a matched character string are arranged in a document, it is unnatural that the degree of similarity is lowered when the character strings up to the latter matched character string are taken as a similar character string. For example, such a rule is against human sensation that, when the input character string is "ABCD", the document 1 contains "ABXXXCD", and the document 2 contains "AB", ""ABXXXCD" is a similar character string in the document 1, "AB" is that in the document 2, and the degree of similarity is higher for "AB". It is unnatural that since the document 1 contains an additional matched character string of "CD", it is evaluated to have a lower degree. It is natural that either the degree of similarity for "ABXXXCD" is higher than "AB", or similar character strings in the document 1 are two of "AB" and "CD".

Now, the process of the second embodiment is described. Referring to the flowchart of Figure 8, in this embodiment, steps 602 - 612 are same as before, and the process for step 614 for indicating the conditions in searching the (i+1)-th valid matched character string is changed as follows:

[Equation 22]

$$s(C, i+1) > e(C, i) - (M-1) \quad \text{(Equation 1)}$$

$$s(D, i+1) > e(D, i) \quad \text{(Equation 2)}$$

and

$$s(D, i+1) - e(D, i) - 1 + \max(e(C, i) - s(C, i+1) + 1, 0) \leq L \quad \text{(Equation 3)}$$

Definition of s (C, i), e (C, i), s (D, i), s (D, i) and the like is the same as above.

Equation 1 means that duplicatively appearing characters are allowed up to M-1 characters, otherwise character strings appearing in the same order as that of characters in the input character string are made valid.

Equation 2 means that valid matched character strings are not overlapped with each other in the document.

Equation 3 means that inserted non-matched characters and duplicatively appearing characters are allowed up to L characters together.

In this embodiment, instead of calculating the ratio of valid matched character strings occupying each of the search character string and the similar character strings in the document, and selecting the smaller one as the similarity factor as in the previous embodiment, the similarity factor is calculated by giving marks to similar character strings and dividing them, with the full mark (mark when it is completely matched). The mark for similar character string is calculated by giving a mark to each character under the following rule, and adding them. Accordingly, the process in step 620 of Figure 8 becomes as follows.

Character belonging to the first valid matched character string	1 point
Character belonging to the i-th (i > 1) valid matched character string, and	
position in search character string _† e (C, i-1) + 1 (Equation 4)	1 point
position in search character string _... e (C, i-1) (Equation 5)	-1/(2*L) point
Character not belonging to valid matched character string	-1/L point

Also in this embodiment, when the i-th similar character string has been determined, the (i+1)-th similar character string is determined by starting comparison from the first character after the top character in the i-th similar character string and not belonging to the valid matched character strings constituting the i-th similar character string.

The negative point for the character not belonging to the valid matched character string is set by taking into account the equilibrium between "the more number of non-matched characters is inserted, the less similarity one feels" and "too much non-matched characters are inserted, then it cannot be felt to be one character string". The maximum total of negative points for one non-matched character string is 1/L*L = 1, and the minimum positive point is N ≥ 1 when taking in the next matched character string (2 is particularly recommended for Japanese) . Thus, the negative points never exceed the positive points. In addition, Equation 5 indicates a duplicatively appearing character, while Equation 4 indicates a simple matched character not a duplicatively appearing character. A case where a character duplicatively appears is accommodated by giving to a character expressed by Equation 5 a negative mark smaller than that for a simple non-matched character.

E12. Example of determination of similar character string and degree of similarity in the second embodiment

An example is shown also for N = 2, and L = 3.

[Table 23]

[Example 5]

	123456
Input character string C:	ABCDEF
	12345678
Part of document D: ...	AB·CD·EF ...

Since the first matched character string is "AB", the first valid matched character string is "AB".

[Equation 23]

$$s(C, 1) = 1 \quad e(C, 1) = 2$$

$$s(D, 1) = 1 \quad e(D, 1) = 2$$

According to Equations 1, 2 and 3, the second valid matched character string is "CD".

[Equation 24]

$$s(C, 2) = 3 \quad e(C, 2) = 4$$

$$s(D, 2) = 4 \quad e(D, 2) = 5$$

According to Equations 1, 2 and 3, the third valid matched character string is "EF".

[Equation 25]

s (C, 3) = 5 e (C, 3) = 6

s (D, 3) = 7 e (D, 3) = 8

Since the end of the input character string is reached, the valid matched character strings are three.

[Table 24]

C:	AB		CD		EF	
	1		2		3	
D:	AB	·	CD	·	EF	
	1		2		3	
Points	1,1,		1,1,		1,1,	
			-1/3		-1/3	

The similar character string is "AB.CD.EF" from s (D, 1) to e (D, 3). The degree of similarity = ((1*6 + (-1/3)*2)/6) = 0.88

[Table 27]

[Example 7]

	1234
Input character string C:	ABCD
	1234567891011121314
Part of document D: ...	ABXXXXCDXX X X X X . ...

Since the first matched character string is "AB", the first valid matched character string is "AB". Since the next matched character string "CD" fails to satisfy Equation 3, the valid matched character string is only the first one.

[Table 28]

C:	ABCD	
	1	
D:	ABXXXXCDXXXXXX.	
	1	

The similar character string is "AB". The degree of similarity = 2/4 = 0.5
The first non-valid matched character after "A" is "X". The second similar character string is searched after "X".

[Table 29]

```
C:  ABCD
      1

D:  ABXXXXCDXXXXXX.
          1
```

Thus, the second similar character string is "CD".

[Table 30]

[Example 8]

```
                                  1234567
Input character string C:         ABCDEFG
                                  12345678
Part of document D:          ... ABCCDEFG ...
```

The valid matched character strings are two of "ABC" and "CDEFG".

[Table 31]

```
C:  ABCDEFG
     1 _____
         2

D:  ABC     CDEFG
    1       2
    1,1,1,  1,1,1,1
          -1/6
```

The similar character string is "ABCCDEFG", and the second "C" satisfies Equation 5. Thus, the degree of similarity = ((1*7 + (-1/6)*1)/7) = 0.97.

While the ambiguity search according to the second one of the preferred embodiments of the present invention has been described for the calculation of a similarity factor for fixed length chains, it will be easily understood by those skilled in the art that it can be applied to a character string containing variable length chains.

Claims

1. A method for identifying a unique character string contained in an input document in a computer system, said computer system being able to search at least one stored comparison document, the method comprising the steps of:

   (a) associating and managing position information for a position in said comparison document where a partial comparison document character string extracted from said comparison document exists with said partial comparison document character string;

(b) extracting a partial input character string from said input document, and determining such as a candidate character string;

(c) identifying a partial comparison document character sting which matches part of said candidate character string with a predetermined similarity factor or higher;

(d) identifying position information associated with said partial comparison document character string which matches with said predetermined similarity factor or higher; and

(e) recognizing said candidate character string as the unique character string by comparing appearance frequency information on a part of said candidate character string appearing in said input document with the position information, and evaluating the amount of feature of said candidate character string.

2. A method for searching a document which has a character string similar to a partial input character string existing in an input document in a computer from a plurality of documents to be searched stored in the computer, the method comprising the steps of:

(a) extracting a partial character string from said input document, and determining such as a candidate character string;

(b) evaluating the amount of feature of said candidate character string through comparison between appearance frequency information on a part of said candidate character string appearing in said input document and appearance frequency information on a part of said candidate character string appearing in a comparison document to recognize said candidate character string as a unique character string; and

(c) searching the comparison document having a character string similar to said unique character string from said plurality of document to be searched.

3. A method for identifying a unique character string contained in an input document in a computer system, said computer system being able to search at least one stored comparison document, the method comprising the steps of:

(a) extracting a partial input character string from said input document, and determining such as a candidate character string; and

(b) evaluating the amount of feature of said candidate character string through comparison between appearance frequency information on a part of said candidate character string appearing in said input document and appearance frequency information on a part of said candidate character string appearing in said comparison document to recognize said candidate character string as a unique character string.

4. A method for evaluating similarity between a comparison document and an input document which contains a first unique character string and a second unique cnaracter string input in a computer, said computer system being able to search a stored comparison document, the method comprising the steps of:

(a) calculating a first weight value corresponding to said first unique character string from appearance frequency information on a part of said first unique character string appearing in said input document;

(b) calculating a second weight value corresponding to said second unique character string from appearance frequency information on a part of said second unique character string appearing in the input document;

(c) calculating a first appearance frequency value on a part of said first unique character string appearing in said comparison document;

(d) calculating a second appearance frequency value on a part of said second unique character string appearing in said comparison document; and

(e) calculating the similarity factor of said comparison document from the first appearance frequency value taking said first weight value into account and the second appearance frequency value taking said second

weight value into account.

5. An apparatus for identifying a unique character string contained in an input document in a computer system, said computer system containing at least one stored comparison document, the apparatus comprising:

(a) a storage device for storing a position information file which associates and manages position information for a position in said comparison document where a partial comparison document character string extracted from said comparison document exists with said partial comparison document character string;

(b) means for extracting a candidate character string from said input document;

(c) means for identifying a partial comparison document character string which matches part of said candidate character string with a predetermined similarity factor or higher;

(d) means for identifying position information which is associated with said partial comparison document character string with the predetermined similarity factor or higher in said position information file; and

(e) means for recognizing said candidate character string as tne unique character string by comparing appearance frequency information on a part of said candidate character string appearing in said input document with said position information, and evaluating the amount of feature of said candidate character string.

6. An apparatus for searching a document having a character string similar to a partial input character string which exists in an input document in a computer from a plurality of documents to be searched which are stored in the computer, the apparatus comprising:

(a) an input device for identifying said input document and instructing execution of search;

(b) means for detecting from said input device the fact that said input document is identified and that said instruction of search is input;

(c) means for extracting a candidate character string from said input document in response to the detection of the fact that said input document is identified and that said instruction of search is input;

(d) means for calculating the amount of feature through comparison between appearance frequency information on a part of said candidate character string appearing in said input document and appearance frequency information on a part of said candidate character string appearing in a comparison document;

(e) means for determining said candidate character string as a unique character string by evaluating said amount of feature;

(f) means for searching the document to be searched having a character string similar to said unique character string from a plurality of documents to be searched; and

(g) a display device for displaying the document to be searched having à character string similar to said unique character string.

7. An apparatus for identifying a unique character string contained in an input document in a computer system, said computer system containing at least one stored comparison document, the apparatus comprising:

(a) means for extracting a candidate character string from said input document; and

(b) means for determining said candidate character string as a unique cnaracter string by evaluating tne amount or feature of said candidate character string through comparison between appearance frequency information on a part of said candidate character string appearing in said input document and appearance frequency information on a part of said candidate character string appearing in said comparison document.

8. An apparatus for evaluating similarity between a comparison document and an input document containing a unique character string in a computer system, said computer system containing a stored comparison document, the ap-

paratus comprising:

(a) means for calculating a weight value corresponding to said unique character string from appearance frequency information on a part of said unique character string appearing in said input document: and

(b) means for calculating the similarity factor of said comparison document from the appearance frequency information on a part of said unique character string appearing in said comparison document and said weight value.

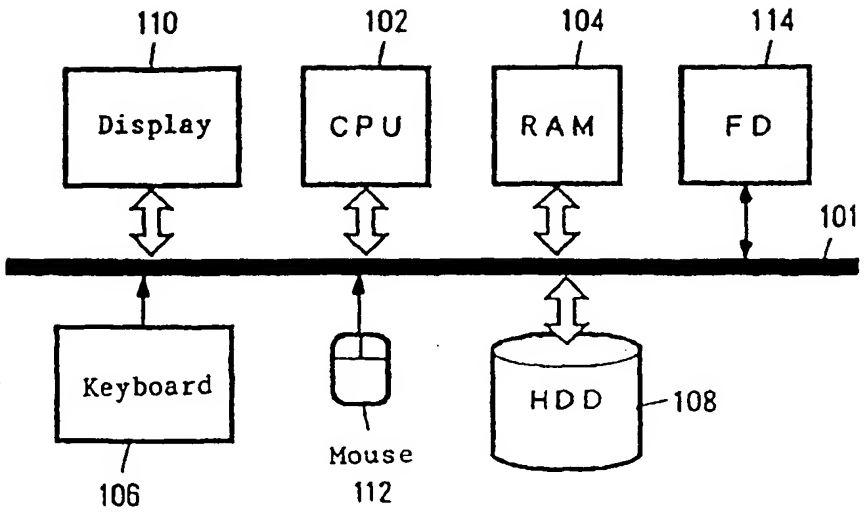
110
102
104
114
Display
CPU
RAM
FD
101
Keyboard
Mouse
HDD
108
106
112

FIG. 1

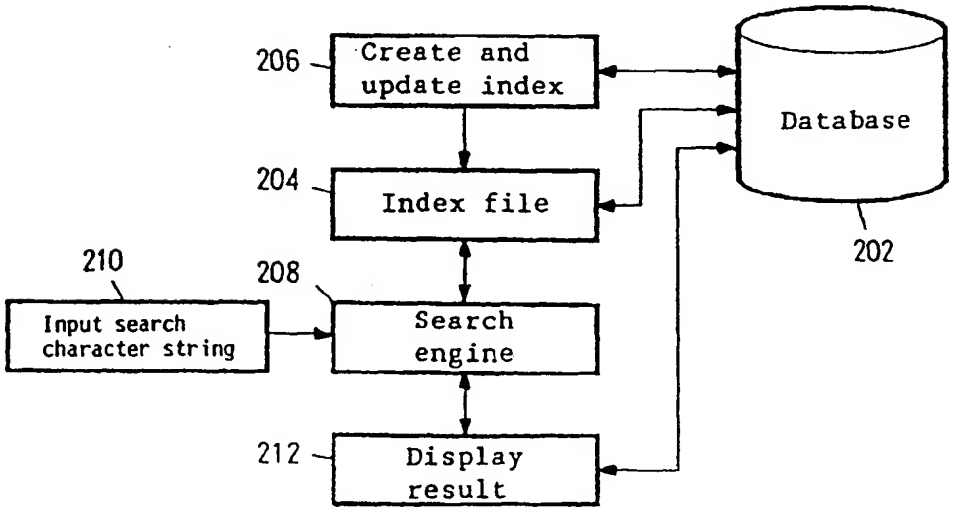
206
Create and update index
Database
204
Index file
202
210
208
Input search character string
Search engine
212
Display result

FIG. 2

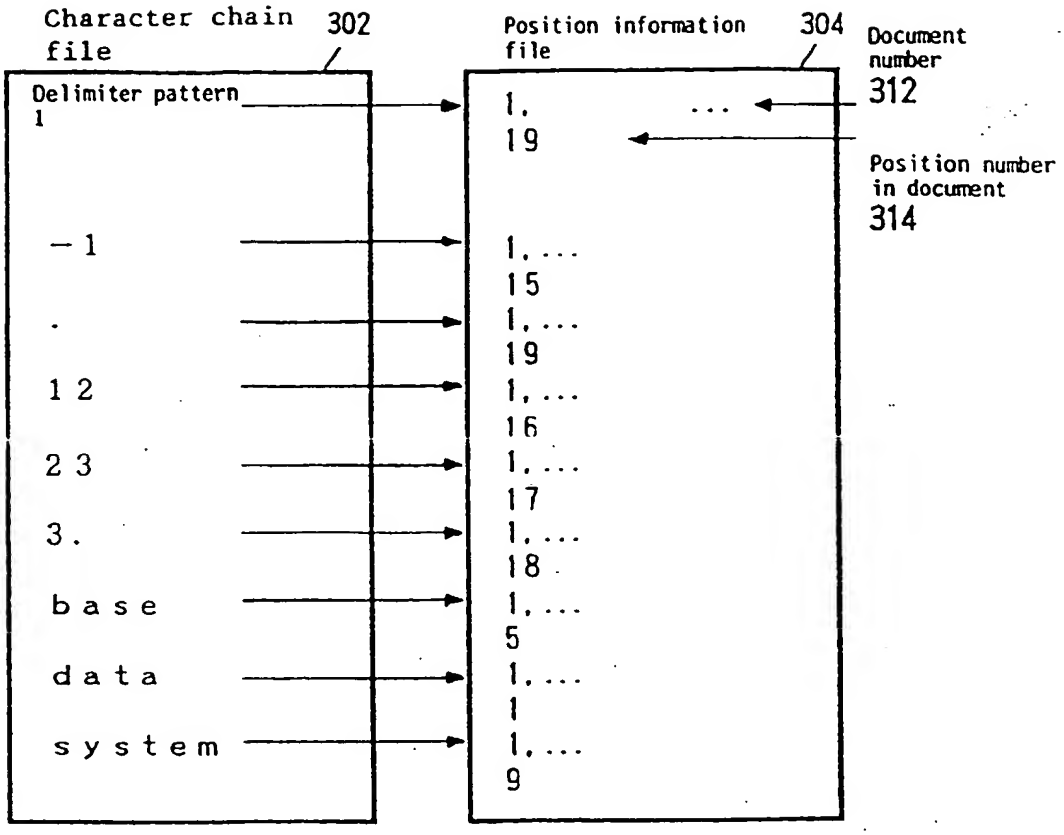
Character chain file
302
Position information file
304
Document number
312
Position number in document
314
Delimiter pattern 1
1.
19
— 1
1, ...
15
.
1, ...
19
1 2
1, ...
16
2 3
1, ...
17
3.
1, ...
18
base
1, ...
5
data
1, ...
1
system
1, ...
9

FIG. 3

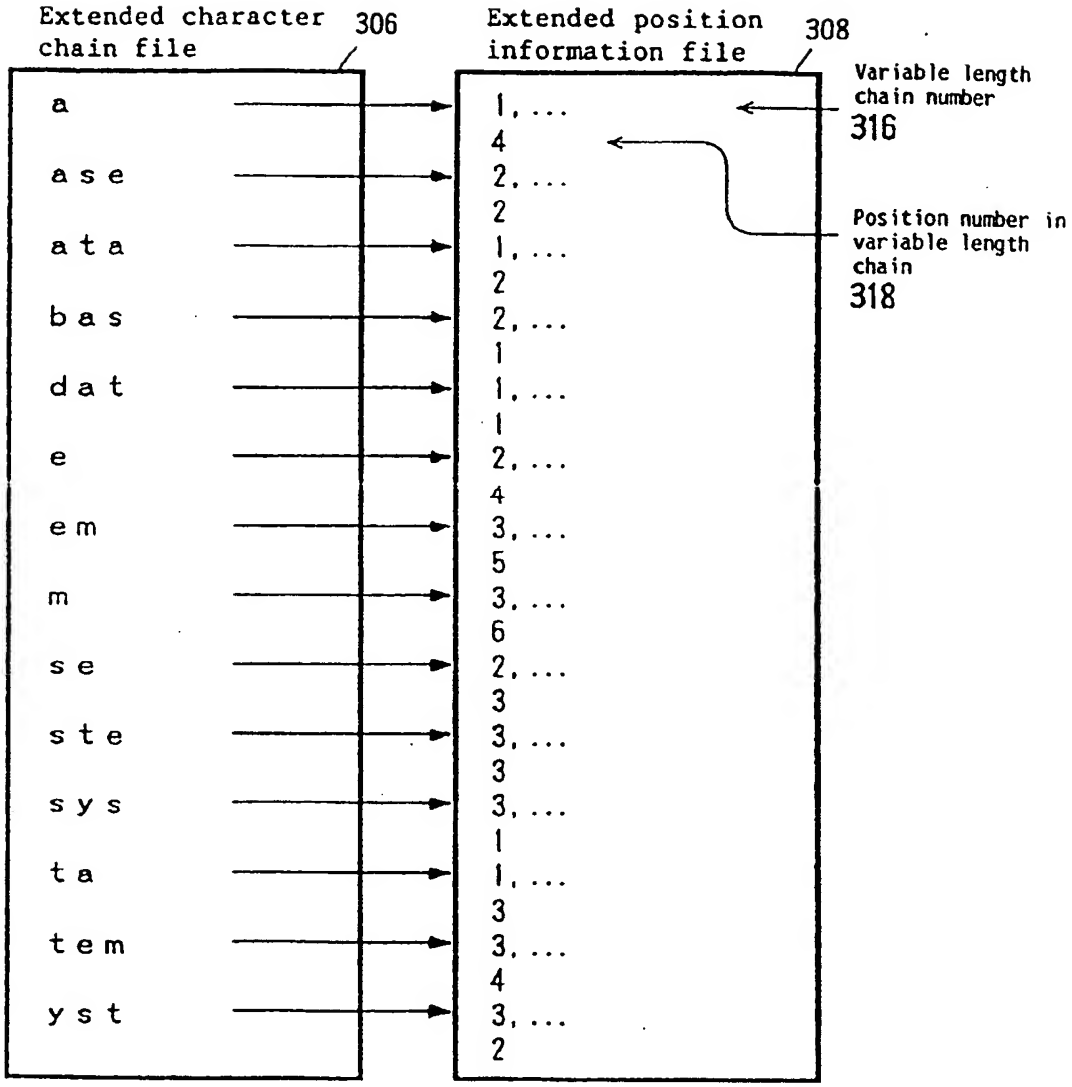
Extended character chain file
306
Extended position information file
308
Variable length chain number
316
Position number in variable length chain
318
a
ase
ata
bas
dat
e
em
m
se
ste
sys
ta
tem
yst
1, ...
4
2, ...
2
1, ...
2
2, ...
1
1, ...
1
2, ...
4
3, ...
5
3, ...
6
2, ...
3
3, ...
3
3, ...
1
1, ...
3
3, ...
4
3, ...
2

FIG. 4

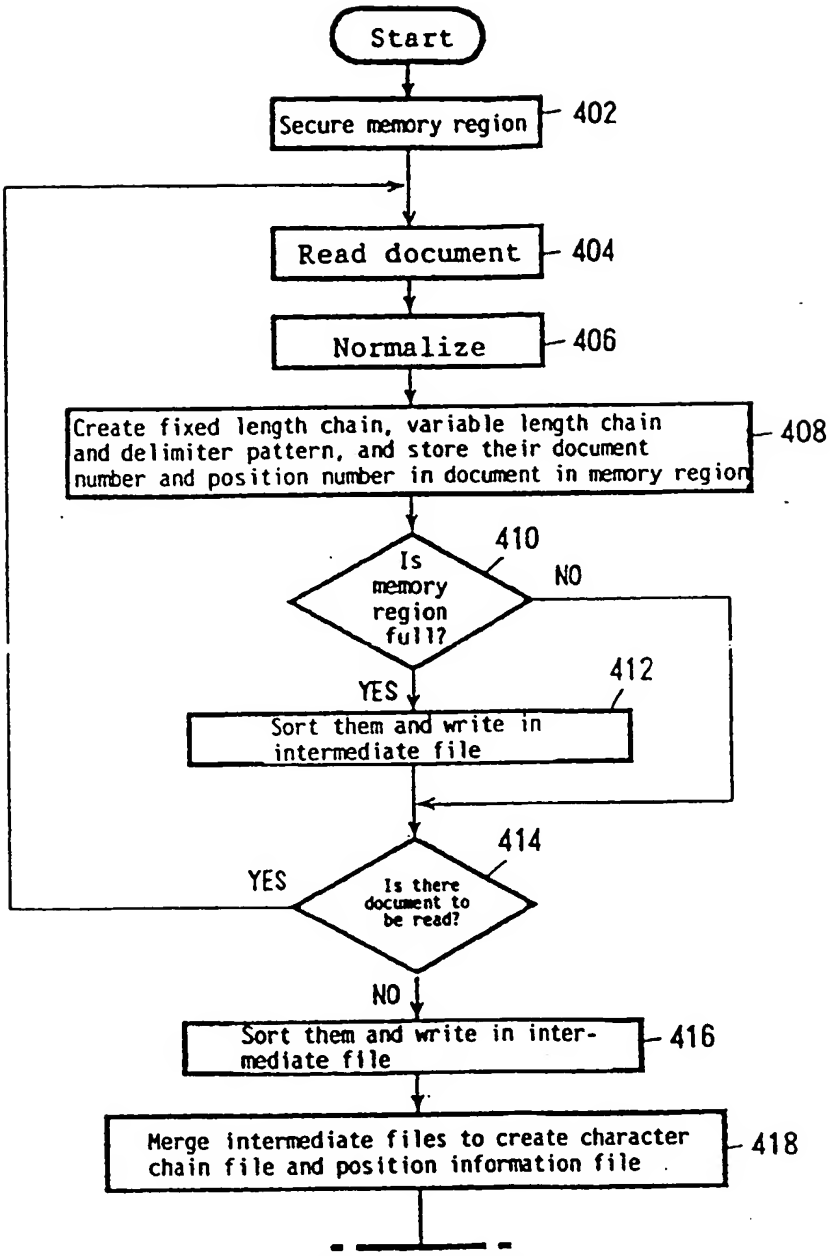
Start
Secure memory region
402
Read document
404
Normalize
406
Create fixed length chain, variable length chain and delimiter pattern, and store their document number and position number in document in memory region
408
Is memory region full?
410
NO
YES
412
Sort them and write in intermediate file
Is there document to be read?
414
YES
NO
Sort them and write in intermediate file
416
Merge intermediate files to create character chain file and position information file
418
(To Figure 6)

FIG. 5

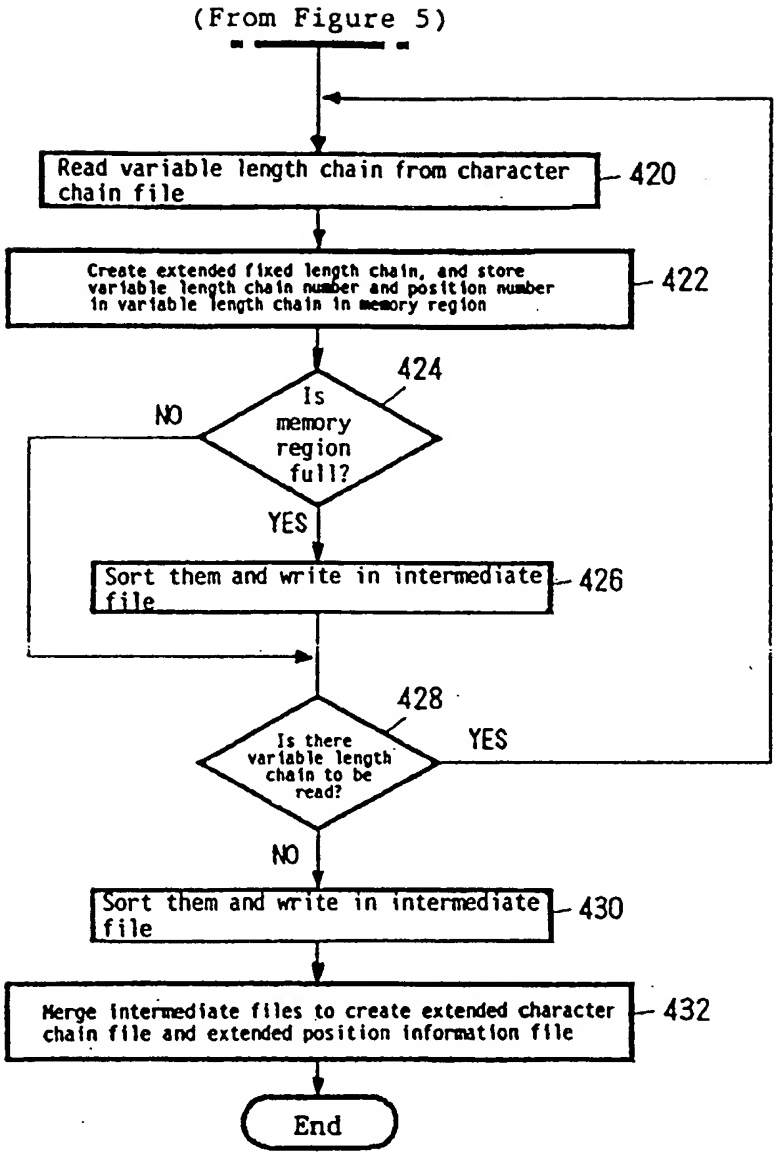
(From Figure 5)
Read variable length chain from character chain file
420
Create extended fixed length chain, and store variable length chain number and position number in variable length chain in memory region
422
424
Is memory region full?
NO
YES
Sort them and write in intermediate file
426
428
Is there variable length chain to be read?
YES
NO
Sort them and write in intermediate file
430
Merge intermediate files to create extended character chain file and extended position information file
432
End

FIG. 6

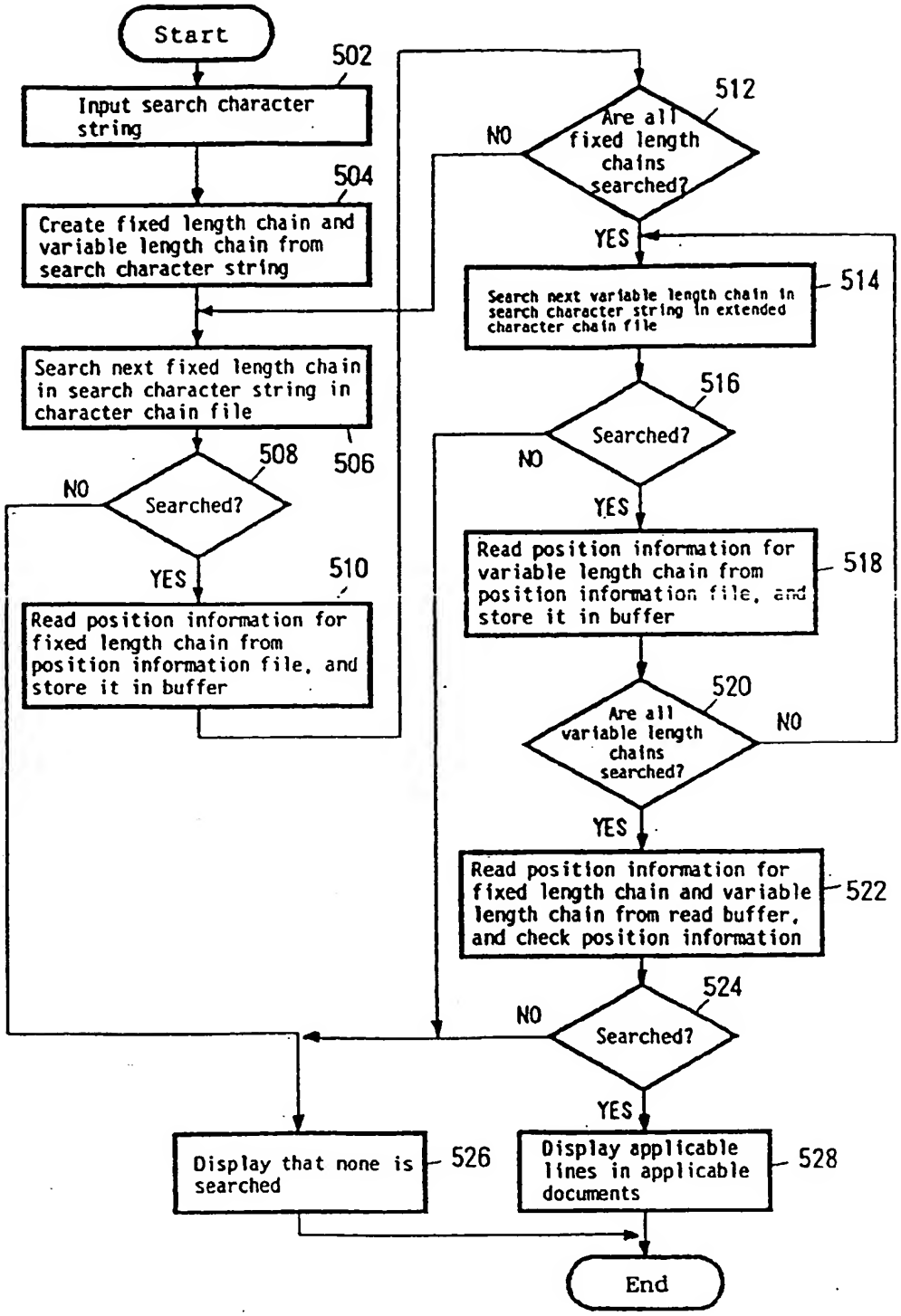
Start
502
Input search character string
504
Create fixed length chain and variable length chain from search character string
Search next fixed length chain in search character string in character chain file
506
508
Searched?
NO
YES
510
Read position information for fixed length chain from position information file, and store it in buffer
512
Are all fixed length chains searched?
NO
YES
514
Search next variable length chain in search character string in extended character chain file
516
Searched?
NO
YES
518
Read position information for variable length chain from position information file, and store it in buffer
520
Are all variable length chains searched?
NO
YES
522
Read position information for fixed length chain and variable length chain from read buffer, and check position information
524
Searched?
NO
YES
526
Display that none is searched
528
Display applicable lines in applicable documents
End

FIG. 7

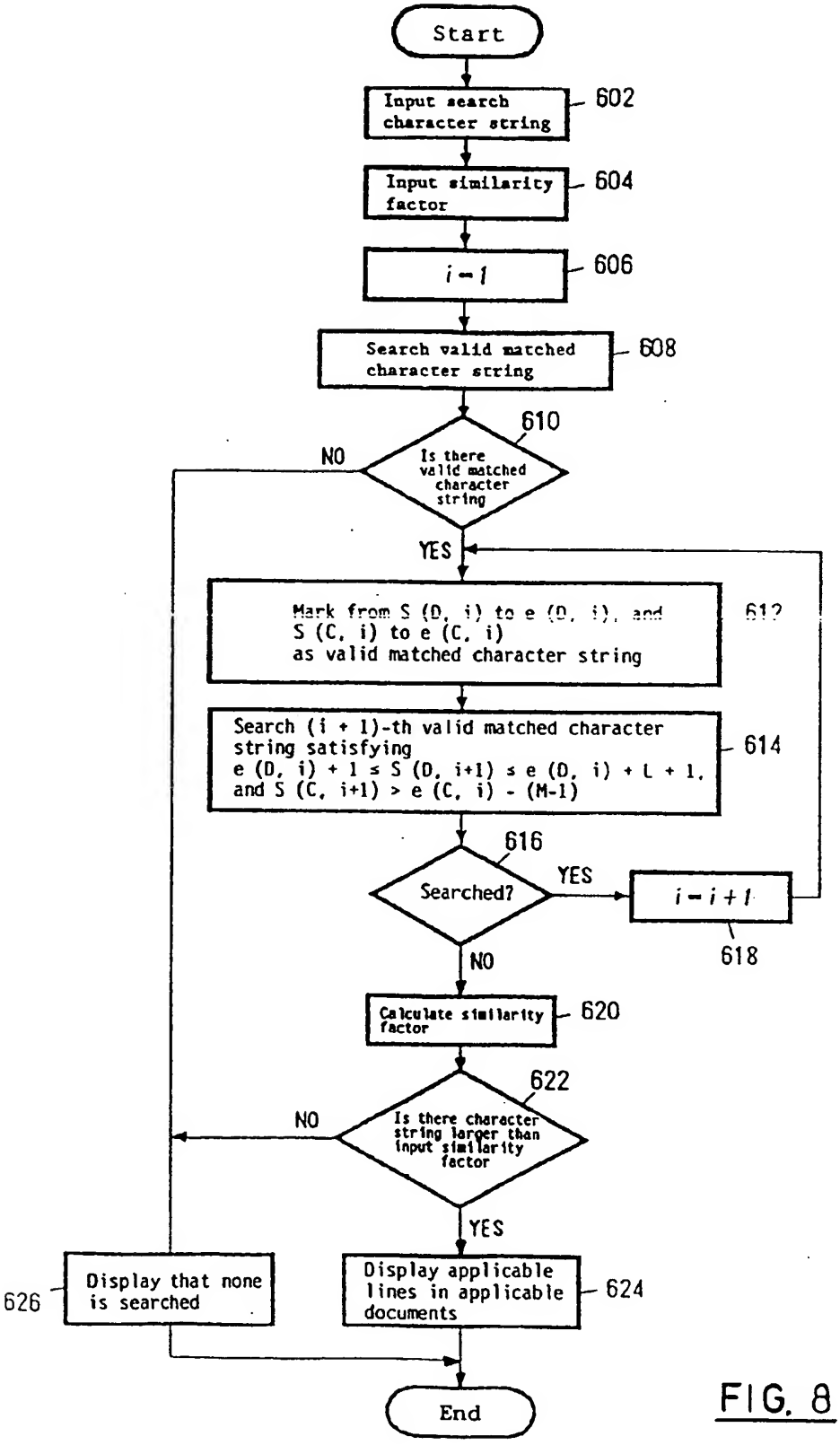
Start
Input search character string
602
Input similarity factor
604
i = 1
606
Search valid matched character string
608
610
Is there valid matched character string
NO
YES
Mark from S (D, i) to e (D, i), and S (C, i) to e (C, i) as valid matched character string
612
Search (i + 1)-th valid matched character string satisfying e (D, i) + 1 ≤ S (D, i+1) ≤ e (D, i) + L + 1, and S (C, i+1) > e (C, i) - (M-1)
614
616
Searched?
YES
i = i + 1
618
NO
Calculate similarity factor
620
622
Is there character string larger than input similarity factor
NO
YES
Display that none is searched
626
Display applicable lines in applicable documents
624
End

FIG. 8

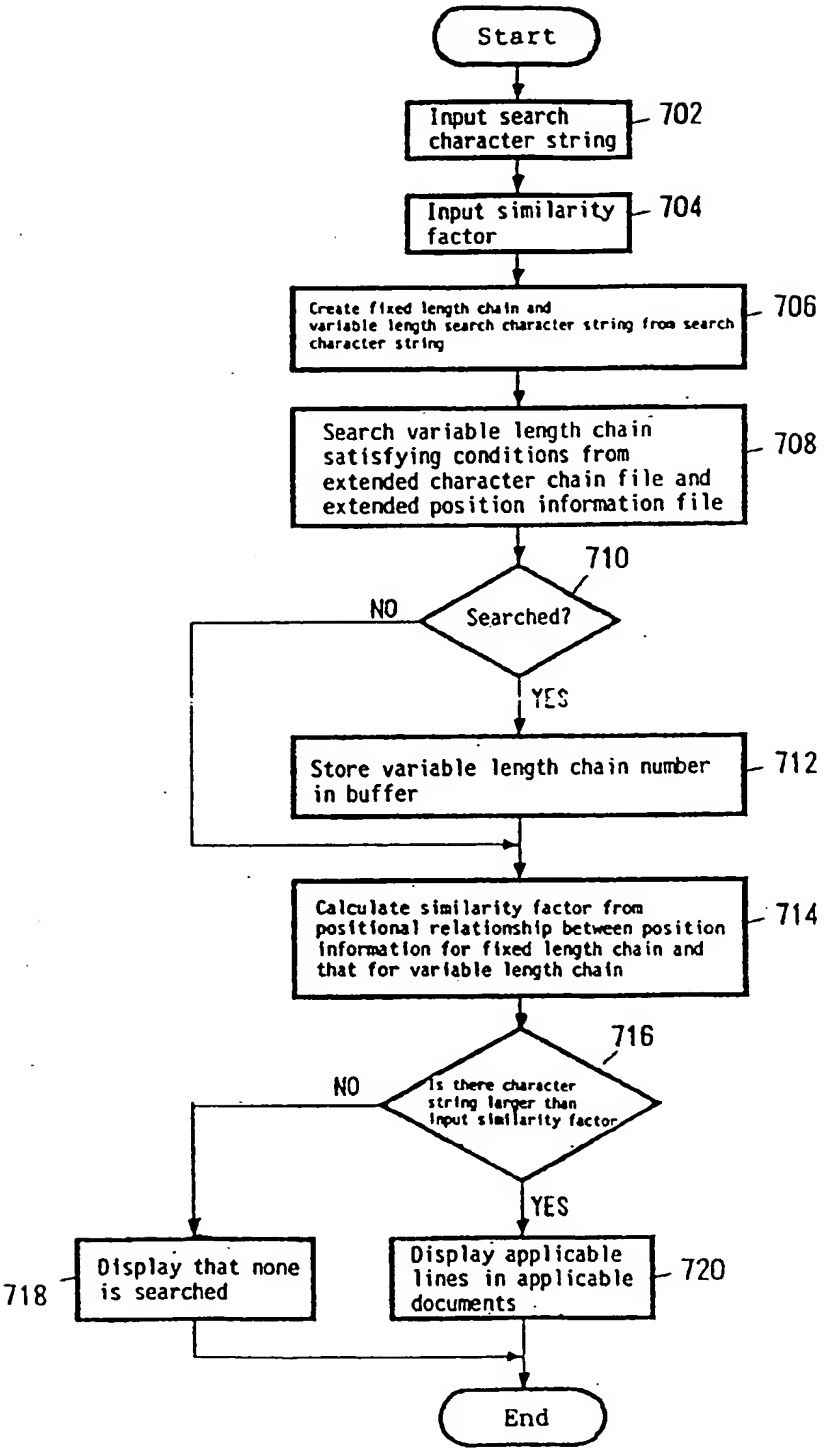
Start
Input search character string
702
Input similarity factor
704
Create fixed length chain and variable length search character string from search character string
706
Search variable length chain satisfying conditions from extended character chain file and extended position information file
708
710
Searched?
NO
YES
Store variable length chain number in buffer
712
Calculate similarity factor from positional relationship between position information for fixed length chain and that for variable length chain
714
716
Is there character string larger than input similarity factor
NO
YES
Display that none is searched
718
Display applicable lines in applicable documents
720
End

FIG. 9

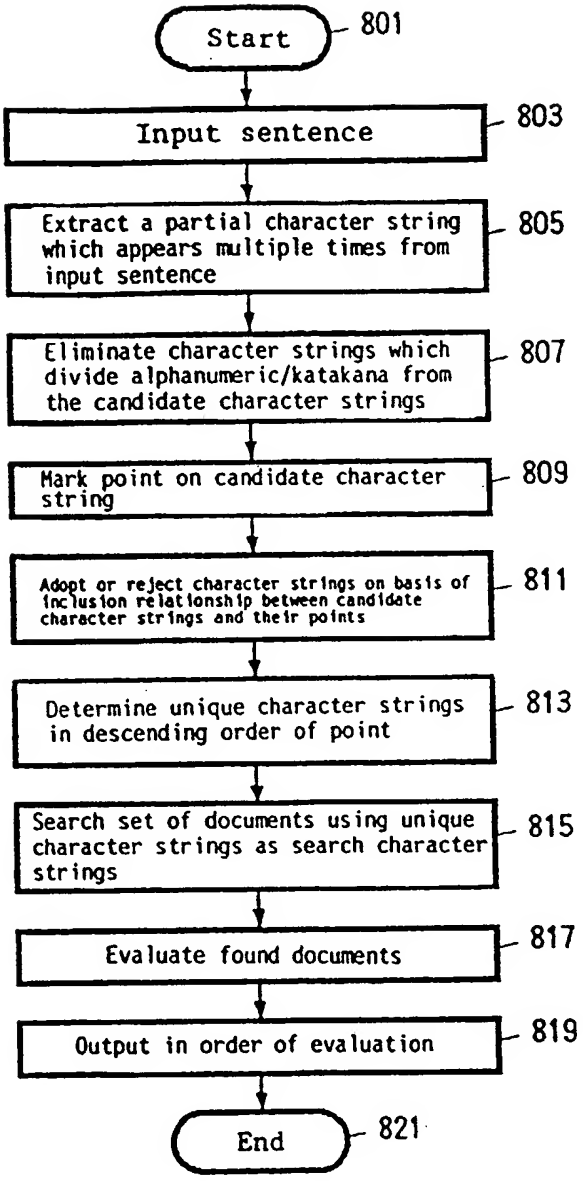
Start
801
Input sentence
803
Extract a partial character string which appears multiple times from input sentence
805
Eliminate character strings which divide alphanumeric/katakana from the candidate character strings
807
Mark point on candidate character string
809
Adopt or reject character strings on basis of inclusion relationship between candidate character strings and their points
811
Determine unique character strings in descending order of point
813
Search set of documents using unique character strings as search character strings
815
Evaluate found documents
817
Output in order of evaluation
819
End
821

FIG. 10

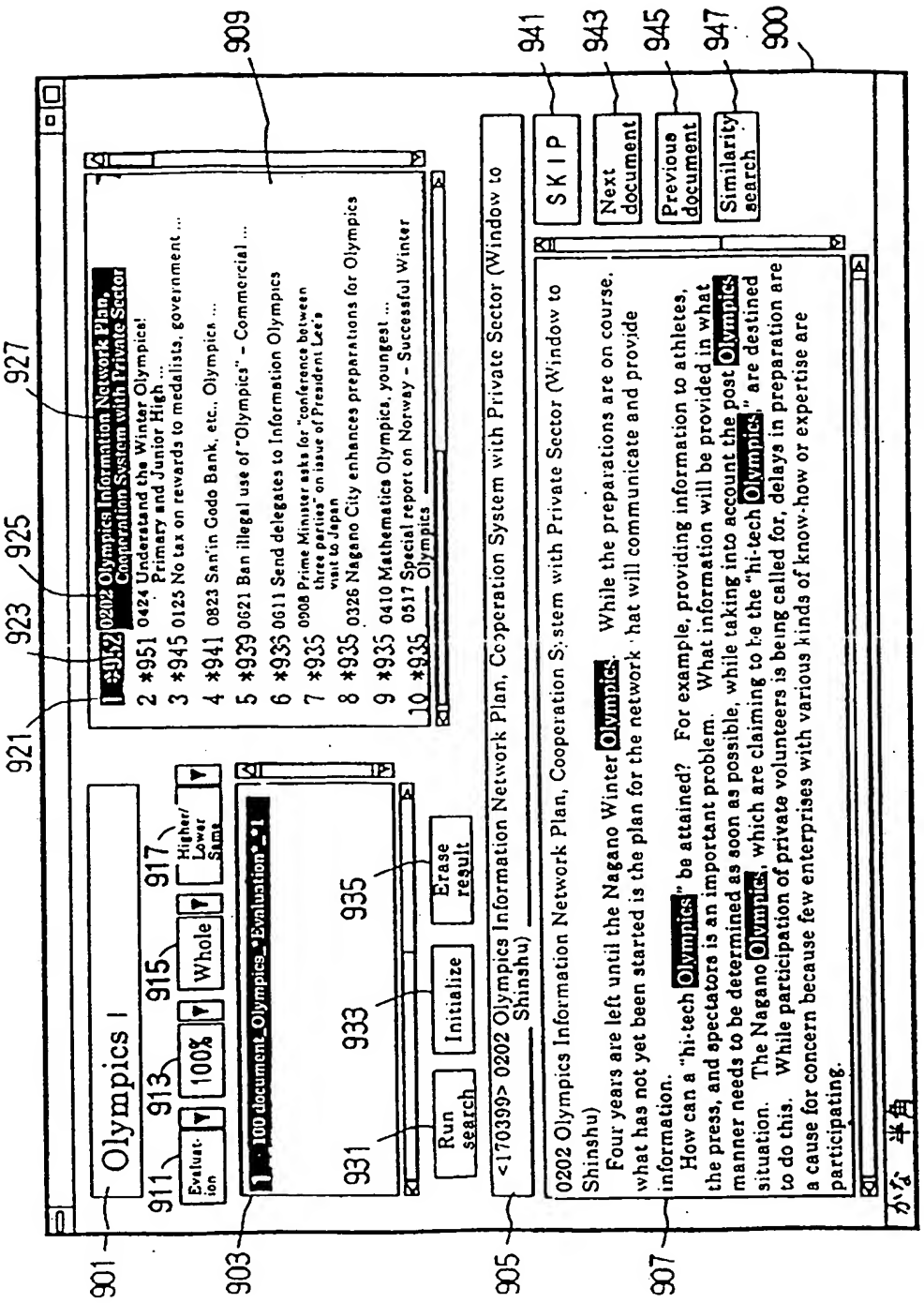
Olympics I
Evaluation
100%
Whole
Higher/ Lower Sims
100 document_Olympics_"Evaluation"_"1
Run search
Initialize
Erase result
0202 Olympics Information Network Plan, Cooperation System with Private Sector
2 *951 0424 Underatand the Winter Olympics!
3 *945 0125 No tax on rewards to medalists, government ...
4 *941 0823 San'in Godo Bank, etc., Olympics ...
5 *939 0621 Ban illegal use of "Olympics" - Commercial ...
6 *935 0611 Send delegates to Information Olympics
7 *935 0908 Prime Minister asks for "conference between three parties" on issue of President Lee's visit to Japan
8 *935 0326 Nagano City enhances preparations for Olympics
9 *935 0410 Mathematics Olympics, youngest ...
10 *935 0517 Special report on Norway - Successful Winter Olympics
<170399> 0202 Olympics Information Network Plan, Cooperation System with Private Sector (Window to Shinshu)
0202 Olympics Information Network Plan, Cooperation System with Private Sector (Window to Shinshu)
Four years are left until the Nagano Winter Olympics. While the preparations are on course, what has not yet been started is the plan for the network that will communicate and provide information.
How can a "hi-tech Olympics" be attained? For example, providing information to athletes, the press, and spectators is an important problem. What information will be provided in what manner needs to be determined as soon as possible, while taking into account the post Olympics situation. The Nagano Olympics, which are claiming to be the "hi-tech Olympics," are destined to do this. While participation of private volunteers is being called for, delays in preparation are a cause for concern because few enterprises with various kinds of know-how or expertise are participating.
SKIP
Next document
Previous document
Similarity search
901
903
905
907
909
911
913
915
917
921
923
925
927
931
933
935
941
943
945
947
900

FIG. 11

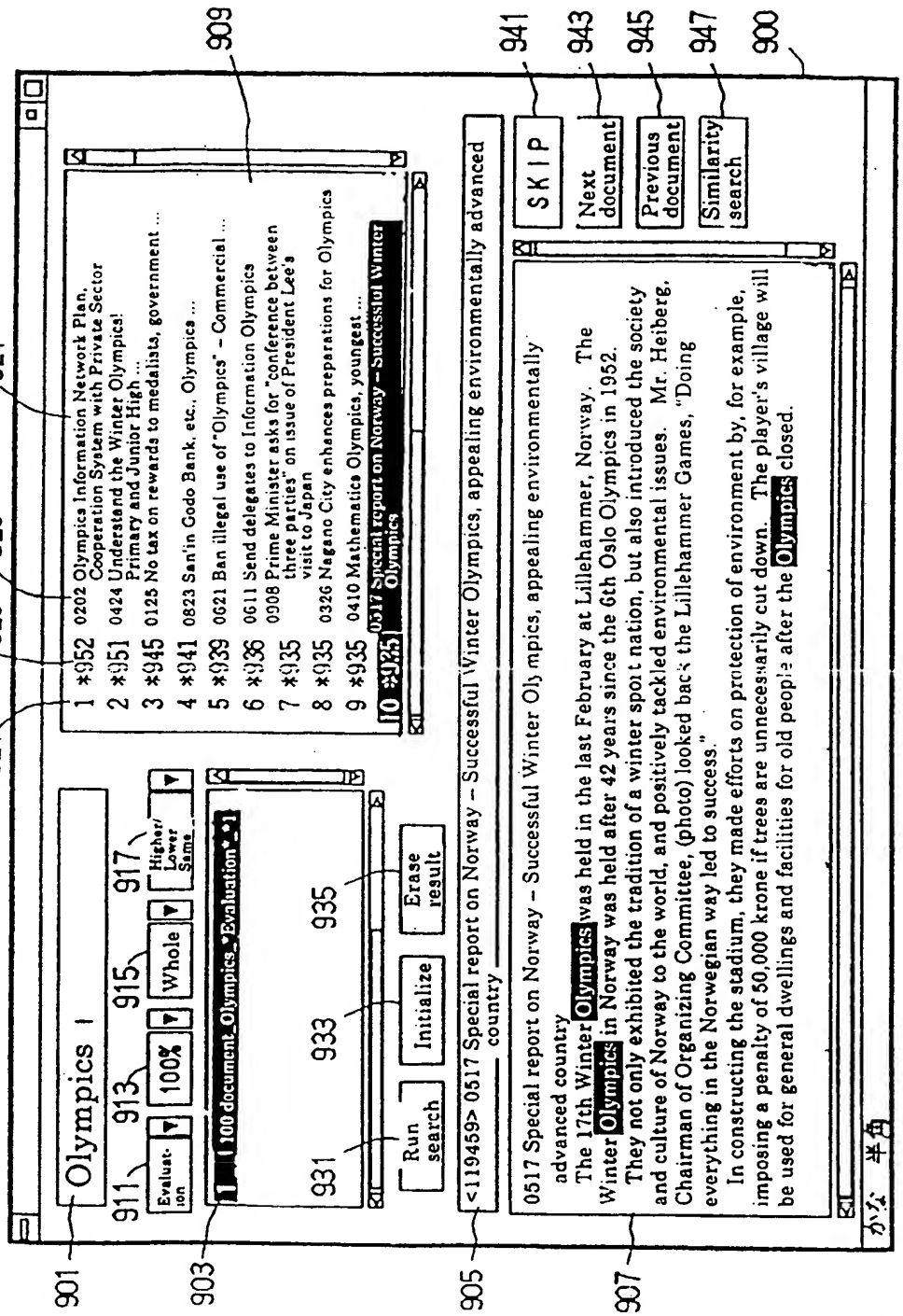
Olympics !
Evaluation
100%
Whole
Higher/ Lower Same
Run search
Initialize
Erase result
1 100 document Olympics "Evaluation" *1
1 *952 0202 Olympics Information Network Plan, Cooperation System with Private Sector
2 *951 0424 Understand the Winter Olympics! Primary and Junior High ...
3 *945 0125 No tax on rewards to medalists, government ...
4 *941 0823 San'in Godo Bank, etc., Olympics ...
5 *939 0621 Ban illegal use of "Olympics" - Commercial ...
6 *936 0611 Send delegates to Information Olympics
7 *935 0908 Prime Minister asks for "conference between three parties" on issue of President Lee's visit to Japan
8 *935 0326 Nagano City enhances preparations for Olympics
9 *935 0410 Mathematics Olympics, youngest ...
10 *935 0517 Special report on Norway – Successful Winter Olympics
<119459> 0517 Special report on Norway – Successful Winter Olympics, appealing environmentally advanced country
0517 Special report on Norway – Successful Winter Olympics, appealing environmentally advanced country
The 17th Winter Olympics was held in the last February at Lillehammer, Norway. The Winter Olympics in Norway was held after 42 years since the 6th Oslo Olympics in 1952.
They not only exhibited the tradition of a winter sport nation, but also introduced the society and culture of Norway to the world, and positively tackled environmental issues. Mr. Heiberg, Chairman of Organizing Committee, (photo) looked back the Lillehammer Games, "Doing everything in the Norwegian way led to success."
In constructing the stadium, they made efforts on protection of environment by, for example, imposing a penalty of 50,000 krone if trees are unnecessarily cut down. The player's village will be used for general dwellings and facilities for old people after the Olympics closed.
SKIP
Next document
Previous document
Similarity search
かな 半角
901
903
905
907
909
911
913
915
917
921
923
925
927
931
933
935
941
943
945
947
900

FIG. 12

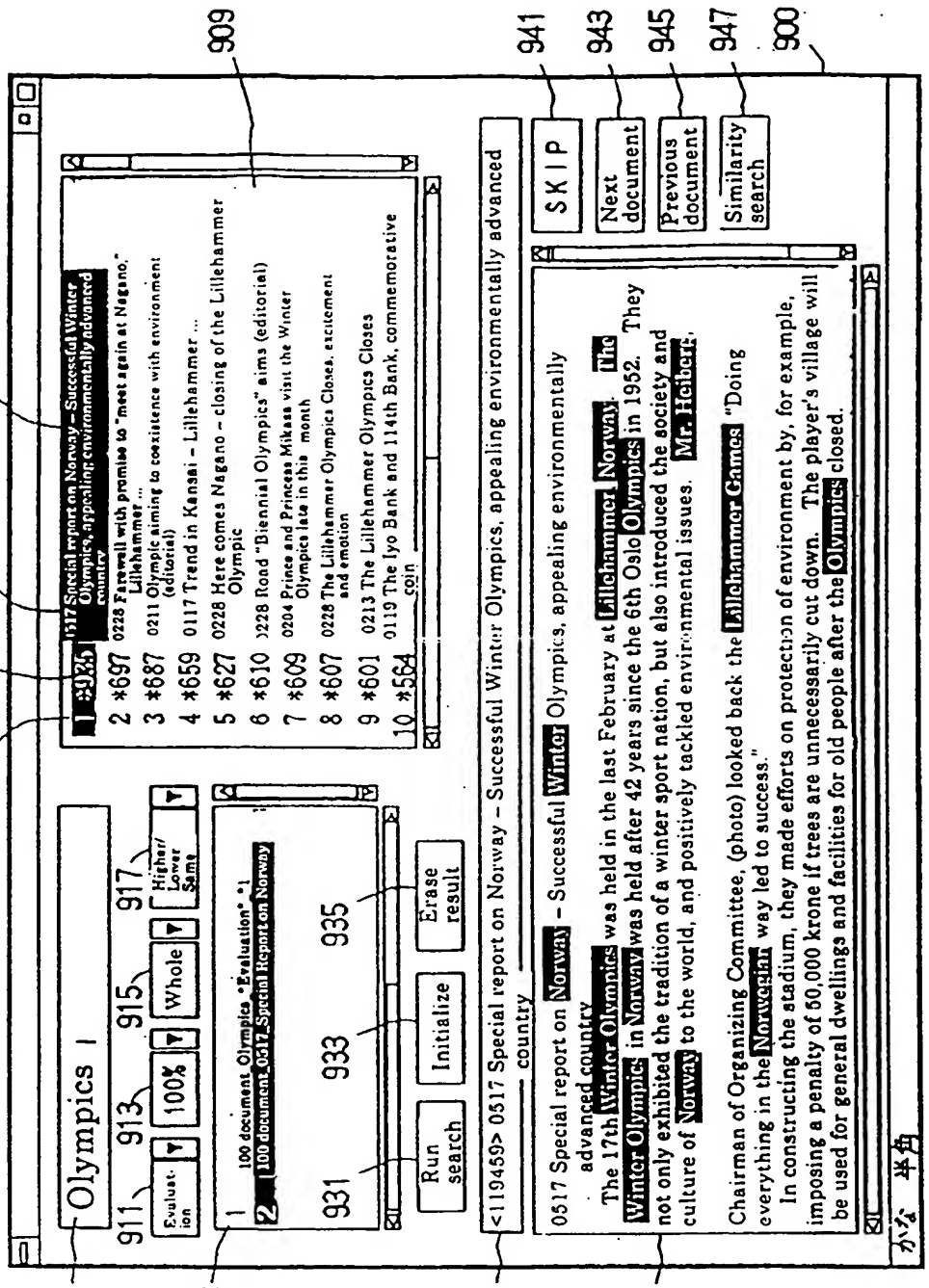
Olympics
Evaluation
100%
Whole
Higher/ Lower Same
911
913
915
917
100 document Olympics "Evaluation" "1
100 document_0517_Special Report on Norway
931
933
935
Run search
Initialize
Erase result
0117 Special report on Norway – Successful Winter Olympics, appealing environmentally advanced country
0228 Farewell with promise to "meet again at Nagano," Lillehammer ...
0211 Olympic aiming to coexistence with environment (editorial)
0117 Trend in Kansai – Lillehammer ...
0228 Here comes Nagano – closing of the Lillehammer Olympic
0228 Road "Biennial Olympics" aims (editorial)
0204 Prince and Princess Mikasa visit the Winter Olympics late in this month
0228 The Lillehammer Olympics Closes, excitement and emotion
0213 The Lillehammer Olympics Closes
0119 The Iyo Bank and 114th Bank, commemorative coin
<119459> 0517 Special report on Norway – Successful Winter Olympics, appealing environmentally advanced country
0517 Special report on Norway – Successful Winter Olympics, appealing environmentally advanced country
The 17th Winter Olympics was held in the last February at Lillehammer, Norway. The Winter Olympics in Norway was held after 42 years since the 6th Oslo Olympics in 1952. They not only exhibited the tradition of a winter sport nation, but also introduced the society and culture of Norway to the world, and positively tackled environmental issues. Mr. Heiberg, Chairman of Organizing Committee, (photo) looked back the Lillehammer Games "Doing everything in the Norwegian way led to success."
In constructing the stadium, they made efforts on protection of environment by, for example, imposing a penalty of 50,000 krone if trees are unnecessarily cut down. The player's village will be used for general dwellings and facilities for old people after the Olympics closed.
SKIP
Next document
Previous document
Similarity search
901
903
905
907
909
921
923
925
927
941
943
945
947
900

FIG. 13

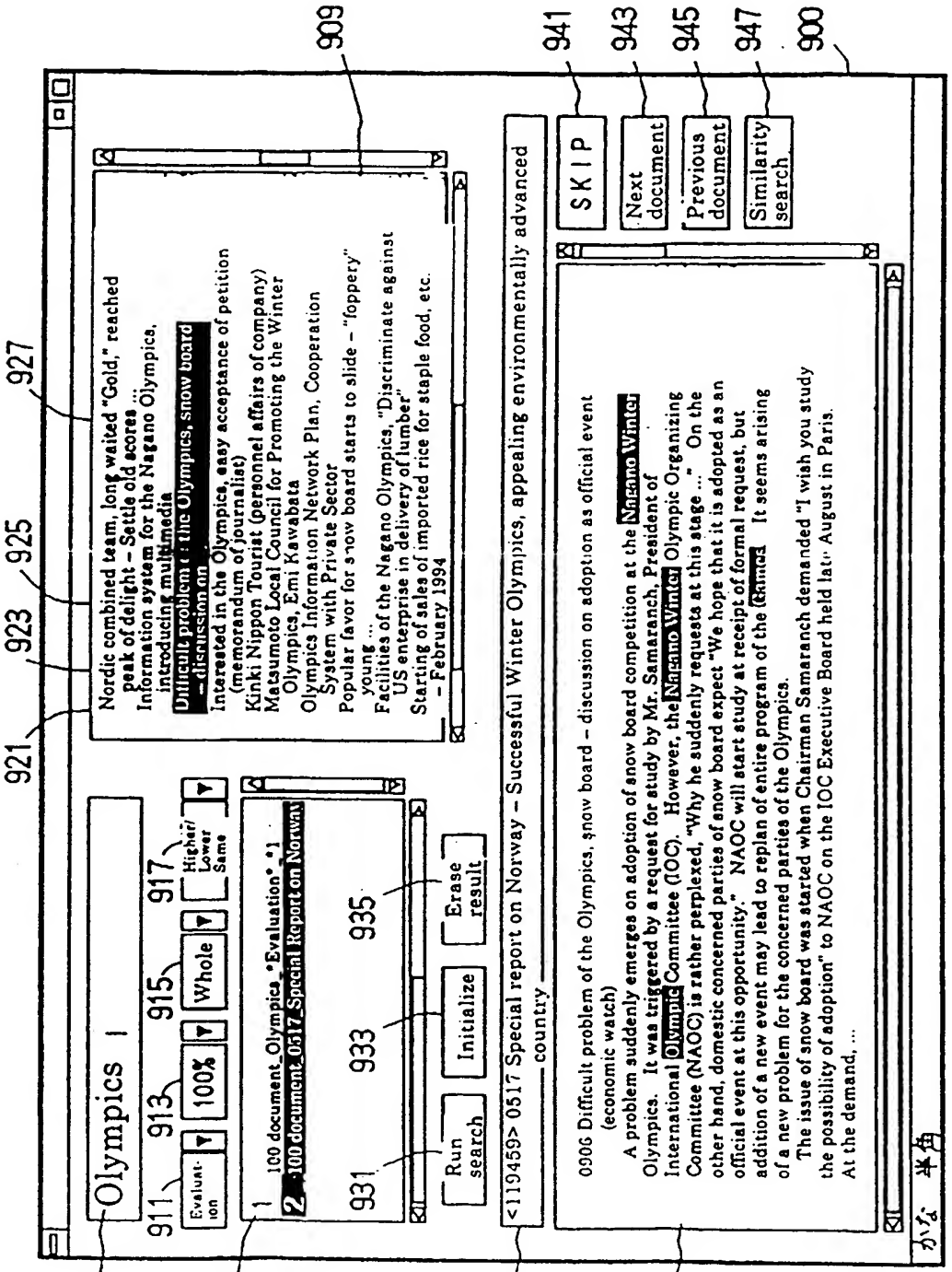
Olympics
Evaluation
100%
Whole
Higher/ Lower Same
Run search
Initialize
Erase result
100 document_Olympics "Evaluation" "1
100 document_0517 Special Report on Norway
Nordic combined team, long waited "Gold," reached peak of delight – Settle old scores ...
Information system for the Nagano Olympics, introducing multimedia
Difficult problem of the Olympics, snow board – discussion on
Interested in the Olympics, easy acceptance of petition (memorandum of journalist)
Kinki Nippon Tourist (personnel affairs of company)
Matsumoto Local Council for Promoting the Winter Olympics, Emi Kawabata
Olympics Information Network Plan, Cooperation System with Private Sector
Popular favor for snow board starts to slide – "foppery" young ...
Facilities of the Nagano Olympics, "Discriminate against US enterprise in delivery of lumber"
Starting of sales of imported rice for staple food, etc. – February 1994
<119459> 0517 Special report on Norway – Successful Winter Olympics, appealing environmentally advanced country
SKIP
Next document
Previous document
Similarity search
0906 Difficult problem of the Olympics, snow board – discussion on adoption as official event (economic watch)
A problem suddenly emerges on adoption of snow board competition at the Nagano Winter Olympics. It was triggered by a request for study by Mr. Samaranch, President of International Olympic Committee (IOC). However, the Nagano Winter Olympic Organizing Committee (NAOC) is rather perplexed, "Why he suddenly requests at this stage ..." On the other hand, domestic concerned parties of snow board expect "We hope that it is adopted as an official event at this opportunity." NAOC will start study at receipt of formal request, but addition of a new event may lead to replan of entire program of the Games. It seems arising of a new problem for the concerned parties of the Olympics.
The issue of snow board was started when Chairman Samaranch demanded "I wish you study the possibility of adoption" to NAOC on the IOC Executive Board held late August in Paris.
At the demand, ...
かな 半角
901
903
905
907
909
911
913
915
917
921
923
925
927
931
933
935
941
943
945
947
900

FIG. 14

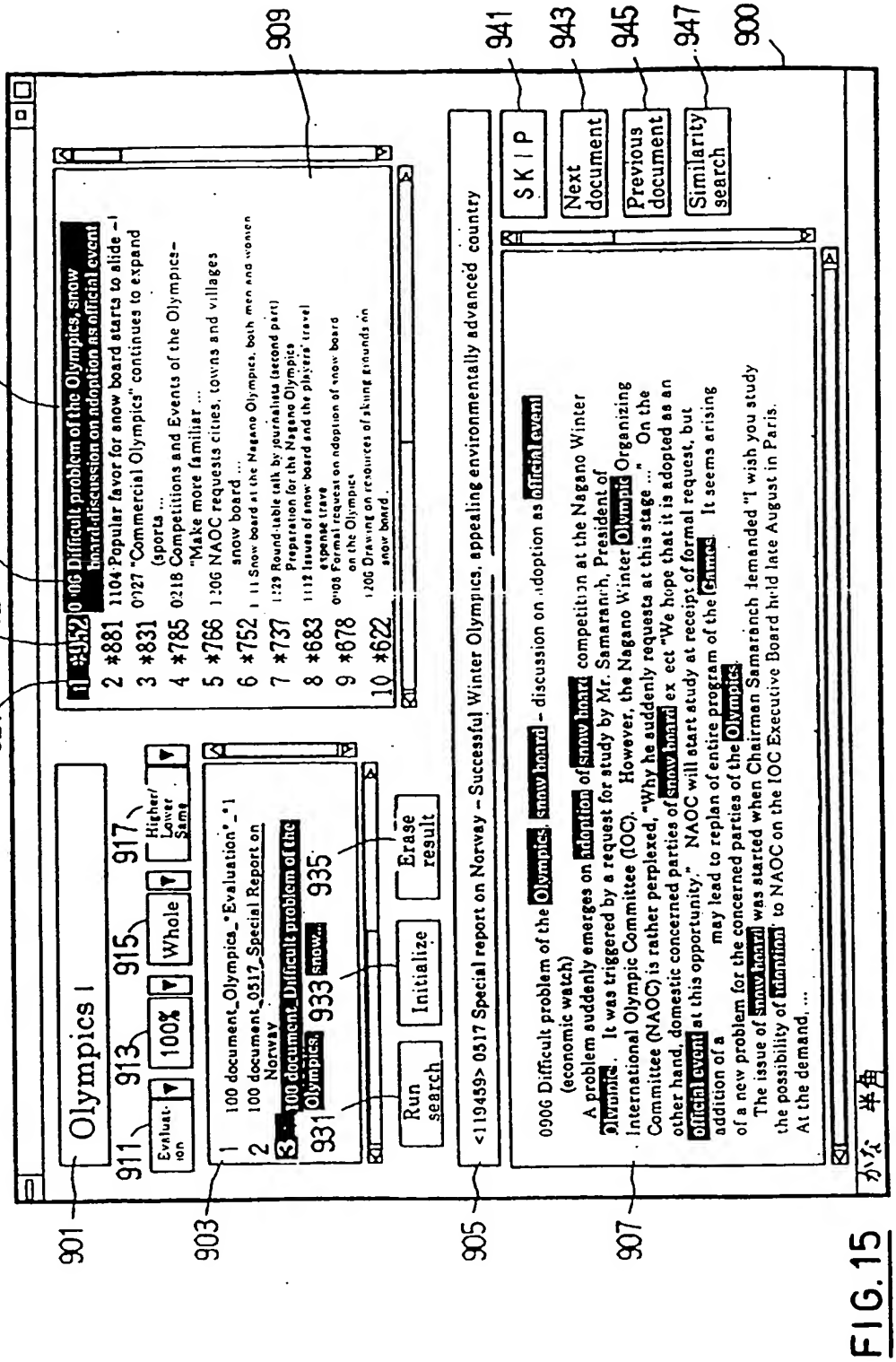
901
Olympics I
911
913
915
917
Evaluation
100%
Whole
Higher/Lower Same
903
1 100 document_Olympics_"Evaluation"_"1
2 100 document_0517_Special Report on Norway
3 100 document_Difficult problem of the Olympics. 933 Snow.. 935
931
Run search
Initialize
Erase result
921
923
925
927
909
1 *952 0:06 Difficult problem of the Olympics, snow board:discussion on adoption as official event
2 *881 1104 Popular favor for snow board starts to slide -1
3 *831 0727 "Commercial Olympics" continues to expand (sports ...
4 *785 0218 Competitions and Events of the Olympics- "Make more familiar ...
5 *766 1206 NAOC requests cities, towns and villages snow board ...
6 *752 1 11 Snow board at the Nagano Olympics, both men and women
7 *737 1229 Round-table talk by journalists (second part) Preparation for the Nagano Olympics
8 *683 1112 Issues of snow board and the players' travel expense trave
9 *678 0908 Formal request on adoption of snow board on the Olympics
10 *622 1206 Drawing on resources of skiing grounds on snow board.
905
<119459> 0517 Special report on Norway -- Successful Winter Olympics, appealing environmentally advanced country
907
0906 Difficult problem of the Olympics, snow board -- discussion on adoption as official event (economic watch)
A problem suddenly emerges on adoption of snow board competition at the Nagano Winter Olympics. It was triggered by a request for study by Mr. Samaranch, President of International Olympic Committee (IOC). However, the Nagano Winter Olympic Organizing Committee (NAOC) is rather perplexed, "Why he suddenly requests at this stage ..." On the other hand, domestic concerned parties of snow board ex ect "We hope that it is adopted as an official event at this opportunity." NAOC will start study at receipt of formal request, but addition of a may lead to replan of entire program of the Games. It seems arising of a new problem for the concerned parties of the Olympics.
The issue of Snow board was started when Chairman Samaranch demanded "I wish you study the possibility of adoption to NAOC on the IOC Executive Board held late August in Paris.
At the demand, ...
941
SKIP
943
Next document
945
Previous document
947
Similarity search
900

FIG. 15

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 30 2600

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	BYTE, vol. 13, no. 5, 1 May 1988, pages 297-312, XP000576194 KIMBRELL R E: "SEARCHING FOR TEXT?. SEND AN N-GRAM" *page 306, box* * page 297, left-hand column, line 1 - page 304, right-hand column, line 15 * ---	1-8	G06F17/30
X	SCIENCE, vol. 267, no. 5199, 10 February 1995, pages 843-848, XP000579827 DAMASHEK M: "GAUGING SIMILARITY WITH N-GRAMS: LANGUAGE-INDEPENDENT CATEGORIZATION OF TEXT" * page 843, column 3, line 1 - page 844, column 2, line 9; figures 1-3 * ---	1-8	
A	EP 0 271 664 A (IBM) 22 June 1988 * page 3, line 6 - page 5, line 12; figure 1 * ---	1-8	TECHNICAL FIELDS SEARCHED (Int.Cl.6)
A	SYSTEMS & COMPUTERS IN JAPAN, vol. 23, no. 2, 1 January 1992, pages 24-38, XP000272460 KIYOHIRO KOBAYASHI ET AL: "A SEARCHING METHOD OF THE MOST SIMILAR STRING IN THE FILE OF A DOCUMENT RETRIEVAL SYSTEM" * page 24, right-hand column, line 1 - page 27, right-hand column, line 6 * ---	1-8	G06F
A	INFORMATION PROCESSING AND MANAGEMENT, vol. 24, no. 5, 1988, GREAT BRITAIN, pages 513-523, XP002035959 SALTON G. ET AL.: "Term-Weighting Approaches in Automatic Text Retrieval" * page 517, line 27 - page 518, line 9; tables 1,2 * -----	1-8	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
BERLIN	24 July 1997	Deane, E

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)